PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-307731
(43)Date of publication of application : 17.11.1998

(51)Int.Cl. G06F 9/46
H04N 7/24

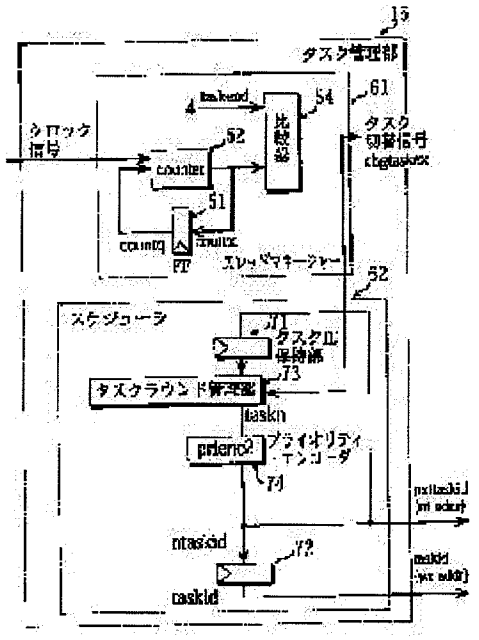
(21)Application number : 10-052073
(22)Date of filing : 04.03.1998

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD
(72)Inventor : TANAKA TAKATOSHI
MAENOBU KIYOSHI
YOSHIOKA KOSUKE
HIRAI MAKOTO
KIYOHARA TOKUZO

(30)Priority
Priority number : 09 49370 Priority date : 04.03.1997 Priority country : JP

(54) PROCESSOR CAPABLE OF EFFICIENTLY EXECUTING ASYNCHRONOUS EVENT TASKS EVEN IN THE CASE EXISTING A LOT OF TASKS TO BE ASYNCHRONOUSLY EXECUTED

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a processor which can unequivocally calculate the optimum lower limit value of operating clock number showing how many operating clocks are to be determined for completing the processing of audio out task and video out task.
SOLUTION: A counter 52 is a counter setting its initial value to '1' and setting its upper limit value to '4' and counts up count values held in a flip-flop 51 like 1, 2, 3, 4, 1, 2, 3, 4 synchronously with a clock signal. Since this clock signal is used for controlling the execution of instructions by an instruction decoding control part, count-up due to the counter 52 is performed each time this instruction decoding control part once performs instruction execution. A comparator 54 compares the value counted up by the counter 52 with an integral value '4' and when the count value at the counter 52 is coincident with the integer value '4', by turning a task switching signal chg-task-ex into high value, switching is performed so as to complete the next task.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平10－307731

(43)公開日　平成10年(1998)11月17日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　9/46	３４０	Ｇ０６Ｆ　9/46	３４０Ｅ
Ｈ０４Ｎ　7/24		Ｈ０４Ｎ　7/13	Ｚ

審査請求　未請求　請求項の数33　ＯＬ　（全 46 頁）

(21)出願番号　特願平10－52073

(22)出願日　平成10年(1998)３月４日

(31)優先権主張番号　特願平9－49370

(32)優先日　平９(1997)３月４日

(33)優先権主張国　日本（ＪＰ）

(71)出願人　000005821
松下電器産業株式会社
大阪府門真市大字門真1006番地

(72)発明者　田中　卓敏
大阪府門真市大字門真1006番地　松下電器産業株式会社内

(72)発明者　前信　潔
大阪府門真市大字門真1006番地　松下電器産業株式会社内

(72)発明者　吉岡　康介
大阪府門真市大字門真1006番地　松下電器産業株式会社内

(74)代理人　弁理士　中島　司朗

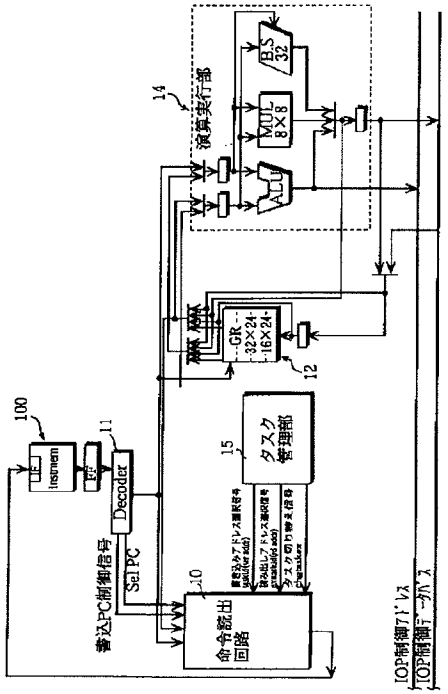
最終頁に続く

(54)【発明の名称】　非同期に実行すべきタスクが多数あっても、非同期イベントタスクを効率良く実行することができるプロセッサ

(57)【要約】

【課題】　オーディオアウトタスク、ビデオアウトタスクの処理を完遂させるには動作クロックをいくらに定めればよいかを示す動作クロック数の最適な下限値が一義的に計算できるプロセッサを提供する。

【解決手段】　カウンタ５２は、初期値が『1』に設定され、上限値が『4』に設定されたカウンタであり、クロック信号に同期してフリップフロップ５１に保持されているカウント値を1,2,3,4,1,2,3,4というようにカウントアップする。このクロック信号は命令解読制御部１１が命令の実行制御のために用いているものであるから、カウンタ５２によるカウントアップは、命令解読制御部１１が命令実行を一回行う度になされる。比較器５４は、カウンタ５２がカウントアップした値と、整数値『４』との比較を行い、もしカウンタ５２のカウント値と整数値『４』とが一致すれば、タスク切替信号chg_task_exをHigh値にして次のタスクを消化するよう切り換えを行う。



【特許請求の範囲】

【請求項１】　n個のタスクを実行対象タスクとしたプロセッサであって、
n個のタスク識別子を出力し、そのタスクに割り当てられた数の命令が実行されると次のタスク識別子を所定の順序で出力する実行タスク指示手段と、
出力されるタスク識別子で特定されるタスク中の実行すべき命令を順次指定する命令指定手段と、
指定された命令を実行する実行手段とを備えることを特徴とするプロセッサ。

【請求項２】　前記実行タスク指示手段は発行された命令をカウントし、カウント値が実行中タスクに割り当てられた命令数になるとタスク切替信号を発生するタスク切替信号発生器と、
タスク切替信号が発される度に次順位のタスク識別子を生成して出力するタスク識別子出力部とを備えることを特徴とする請求項１記載のプロセッサ。

【請求項３】　前記命令指定手段は、
n個のタスクと一対一に対応し、対応するタスクに関して次に読み出すべき先頭アドレスを保持しているｎ個とのアドレスレジスタと、
実行タスク指示手段からタスク識別子が出力されると、その識別子と対応するアドレスレジスタを選択するレジスタ選択部と、
レジスタ選択部で選択されたアドレスレジスタが格納する先頭アドレスを、最初の読出先アドレスとし、タスク識別子が切り替えられるまでの間前記読出先アドレスを順次更新する読出先アドレス生成部と、
次順のタスクに切り換わる際に、読出アドレス生成部で更新された読出アドレスで、切り換え直前までレジスタ選択部で選択されていたアドレスレジスタの格納する先頭アドレスを書き換えるアドレスレジスタ更新部とを備え、
前記実行手段は、前記読出先アドレス生成部が生成したアドレスで特定される命令を実行することを特徴とする請求項２記載のプロセッサ。

【請求項４】　前記タスク切替信号発生器は、
クロックパルスをカウントするカウンタと、
カウンタのカウント値を現在実行中のタスクに割り当てられた命令数と比較し、一致するとタスク切替信号を出力すると共に前記カウンタのリセットを行う比較器とを備えることを特徴とする請求項２記載のプロセッサ。

【請求項５】　前記タスク切替信号発生器は更に、
２種以上の命令数を保持する命令数保持部と、
実行タスク指示手段が指示するタスク識別子によって命令数保持部が保持する一の命令数を選択する命令数選択部とを備え、
選択された命令数が比較器に比較入力として加えられることを特徴とする請求項４記載のプロセッサ。

【請求項６】　前記命令数保持部が保持する命令数には、
第１定数と、第１定数より少ない第２定数があり、
前記命令指定手段は更にn個のタスクにおいて第２定数を割り当てるべきタスクであるペナルティタスクの識別子を保持するペナルティタスク保持部を備え、
前記命令数選択部は実行タスク指示手段がタスク識別子を出力すると、当該タスク識別子がペナルティタスクの識別子であるかを判定する判定部と、
タスク識別子がペナルティタスクを示すと判定されれば、命令数保持部が保持する第２定数を比較器に出力する定数出力部とを備えることを特徴とする請求項５記載のプロセッサ。

【請求項７】　前記タスクには、事象成立の真偽条件の指定を含む事象監視命令が含まれており、
前記実行手段は読出先アドレス生成部が生成したアドレスにて特定される命令を解読する解読手段と、
解読手段による解読結果が事象監視命令であれば、事象監視命令において指定された事象成立の真偽を確認する旨の演算を実行する演算手段とを備え、
前記ペナルティタスク保持部は、
演算手段による演算結果が事象不成立を示すものならば、事象監視命令を含んでいたタスクのタスク識別子をペナルティタスクの識別子として保持することを特徴とする請求項６記載のプロセッサ。

【請求項８】　前記プロセッサは、その内部にバッファと、メモリとを有する情報処理システムに備えられ、
前記プロセッサは、内部バッファ及びメモリにおける状態を示す状態レジスタを複数有し、
前記事象監視命令は、複数の状態レジスタの何れかについての指定と、即値指定とを有し、
前記演算手段は、
解読手段による解読結果が事象監視命令であれば、事象監視命令において指定された状態監視レジスタの保持値と、当該命令において指定された即値とを用いた演算を行うことを特徴とする請求項７記載のプロセッサ。

【請求項９】　前記タスク識別子出力部は各タスク識別子を何番目に出力するかを示す出力順位を、n個の全てのタスクについて記憶する順位記憶部と、
タスク切替信号が発される度に、順位記憶部において出力順位が次順位となるタスク識別子を命令指定手段に出力する選択出力部とを備えることを特徴とする請求項２記載のプロセッサ。

【請求項１０】　前記プロセッサにおいて、
何れかのタスクは緊急扱いを要する旨の緊急宣言命令を含み、
前記実行手段は命令指定手段が指定した命令を解読する解読手段を備え、
前記実行タスク指示手段は解読手段による解読結果が緊急宣言命令であれば、当該命令によって指定されるタスクを緊急扱いを要するタスクとしてその識別子を保持す

る緊急タスクレジスタと、
緊急タスクレジスタが緊急タスクの識別子を保持していない状態においてタスク切替信号が発されれば、選択出力部が出力したタスク識別子を実行手段にスルー出力し、
緊急タスクレジスタが緊急タスクの識別子を保持した状態においてタスク切替信号が発されれば、タスク切替信号のm回の発信につき(m≧2)一回の比率にて、緊急タスクレジスタが保持しているタスク識別子を実行手段に出力する出力比率制御部とを備えることを特徴とする請求項９記載のプロセッサ。
【請求項１１】　前記プロセッサにおいて、
何れかのタスクは緊急扱いを要する旨の緊急宣言命令を含み、
前記実行タスク指示手段は更にプロセッサ外部から入力される所定信号を監視することにより、当該プロセッサ外部が緊急期間内であるか、緊急期間外であるかを判定する監視手段と、
所定信号が緊急期間内である場合に、緊急扱いすべきタスクである緊急タスクの識別子を保持する緊急タスクレジスタと、
緊急期間外においてタスク切替信号が発されれば、選択出力部が未出力のタスクから選んで出力したタスク識別子を実行手段にスルー出力し、
緊急期間内においてタスク切替信号が発されれば、タスク切替信号のm回の発信につき(m≧2)一回の比率にて、緊急タスクレジスタが保持しているタスク識別子を実行手段に出力する出力比率制御部とを備えることを特徴とする請求項９記載のプロセッサ。
【請求項１２】　前記順位記憶部はnビット列を保持するnビットレジスタを備え、
nビット列の各ビットは、それぞれがタスク識別子に対応づけられており、
各タスク識別子の出力順位は、それぞれのタスク識別子に対応づけられたタスクビットが当該nビット列において最上位或は最下位から何ビット目に位置するかにより表現されることを特徴とする請求項１１記載のプロセッサ。
【請求項１３】　前記nビット列において、
タスク識別子が出力済みのタスクに割り当てられたビットは『1』に設定され、未出力タスクに割り当てられたビットは『0』に設定されており、
前記タスク識別子出力部は選択出力部により次順位のタスク識別子が出力されると、nビットレジスタが保持するnビット列のうち、出力されたビットを『1』に設定するビットセット部と、
nビット列の全ビットがビットセット部により『1』に設定されると、nビット列をオールゼロにリセットするビットリセット部とを備え、
前記選択出力部はnビットレジスタが保持するnビット列において『1』から『0』への反転が現れているビット位置に基づいて、次順位に出力すべきタスク識別子を生成すると共に、そのタスク識別子を実行タスク指示手段に出力するプライオリティエンコーダを備えることを特徴とする請求項１２記載のプロセッサ。
【請求項１４】　前記タスク識別子出力部は緊急タスクレジスタが緊急宣言命令を含むタスクのタスク識別子を保持すれば、nビットレジスタが保持しているnビット列において当該緊急タスクに割り当てられたビットを『1』に設定するマスク部を備えることを特徴とする請求項１３記載のプロセッサ。
【請求項１５】　前記プロセッサは、動画ストリームを復号するAVデコードシステムに備えられ、
緊急タスクは、
AVデコードシステムにおける動画ストリームの復号が終ると、復号結果を外部に接続されたディスプレィに出力するタスクであり、
監視手段が監視する所定信号とは、映像信号を表示するためのディスプレィにおける水平帰線期間信号、垂直帰線期間信号であることを特徴とする請求項１４記載のプロセッサ。
【請求項１６】　前記タスク切替信号発生器はクロックパルスをカウントするカウンタと、
カウンタのカウント値を現在実行中のタスクに割り当てられた命令数と比較し、一致するとタスク切替信号を出力すると共に前記カウンタのリセットを行う比較器とを備えることを特徴とする請求項１１記載のプロセッサ。
【請求項１７】　前記タスク切替信号発生器は、
２種以上の命令数を保持する命令数保持部と、
実行タスク指示手段が指示するタスク識別子によって命令数保持部が保持する一の命令数を選択する命令数選択部とを備え、
選択された命令数が比較器に比較入力として加えられることを特徴とする請求項１６記載のプロセッサ。
【請求項１８】　前記命令数保持部が保持する命令数には、
第１定数と、第１定数より多い第２定数があり、
前記タスク識別子出力部は次にタスク識別子が出力された際、第２定数を割り当てるべき救済対象タスク識別子を保持する救済対象タスク保持部を備え、
前記命令数選択部は監視手段が、プロセッサ外部から入力される所定信号が緊急期間内であると判定した場合、実行タスク指示手段がタスク識別子を出力すると、当該タスク識別子が救済対象タスクを示すものかを判定する判定部と、
タスク識別子が救済対象タスクを示すと判定されれば、命令数保持部が保持する第２定数を比較器に出力する定数出力部とを備えることを特徴とする請求項１７記載のプロセッサ。
【請求項１９】　前記プロセッサにおいて何れかのタス

クは何れかのタスクの実行を休眠状態に設定すべき旨の自発休眠命令を含み、
前記実行手段は更に命令指定手段が次に実行すべき命令として指定した命令を解読する解読手段を備え、
前記実行タスク指示手段は更に解読手段による解読結果が自発休眠命令であれば、当該命令により指示されるタスクを休眠扱いを要するタスクとしてそのタスク識別子を保持する休眠タスクレジスタを備え、
前記選択出力部は、
順位記憶部が記憶における次順位のタスク識別子が、休眠タスクのタスク識別子なら、その次順位のタスク識別子を出力することを特徴とする請求項９記載のプロセッサ。
【請求項２０】　何れかのタスクは、他のタスクの休眠状態を解除すべき旨の強制解除命令を含み、
前記実行タスク指示手段は解読手段による解読結果が強制解除命令であれば、休眠タスクレジスタが保持しているタスク識別子を削除する削除部を備えることを特徴とする請求項１９記載のプロセッサ。
【請求項２１】　前記順位記憶部は、
nビット列を保持するnビットレジスタと、
nビット列の各ビットは、それぞれがタスク識別子に対応づけられており、
各タスク識別子の出力順位は、それぞれのタスク識別子に対応づけられたタスクビットが当該nビット列において最上位或は最下位から何ビット目に位置するかにより表現されることを特徴とする請求項２０記載のプロセッサ。
【請求項２２】　前記nビット列において、
タスク識別子が出力済みのタスクに割り当てられたビットは『1』に設定され、未出力タスクに割り当てられたビットは『0』に設定されており、
前記実行タスク出力部は選択出力部により次順位のタスク識別子が出力されると、nビットレジスタが保持するnビット列のうち、出力されたビットを『1』に設定するビットセット部と、
nビット列の全ビットがビットセット部により『1』に設定されると、nビット列をオールゼロにリセットするビットリセット部とを備え、
前記選択出力部はnビットレジスタが保持するnビット列において『1』から『0』への反転が現れているビット位置に基づいて、次順位に出力すべきタスク識別子を生成すると共に、そのタスク識別子を実行タスク指示手段に出力するプライオリティエンコーダを備えることを特徴とする請求項２１記載のプロセッサ。
【請求項２３】　前記実行タスク指示手段は緊急タスクレジスタが自発休眠命令を含むタスクのタスク識別子を保持すれば、nビット列において当該休眠状態タスクに割り当てられたビットを『1』にマスクするマスク部と、
解読手段による解読結果が強制解除命令であれば、nビット列において当該命令にて指定されるタスクのタスク識別子に割り当てられたビットを『0』にしてマスクを解除するアンマスク部とを備えることを特徴とする請求項２２記載のプロセッサ。
【請求項２４】　各タスクに割り当てられたサイクル数を保持するサイクル数保持手段と、
クロックをカウントするカウンタと、
カウンタのカウント値がサイクル数保持手段の最大サイクル数に一致すると、タスク切り替え信号とカウンタリセット信号を発する比較手段と、
予め決められたタスク選択の順序を有し、タスク切替信号が入力されると、タスクの選択を切り替える選択手段と、
選択手段が一のタスクを選択している間、そのタスクの命令を順次指定する命令指定手段と、
選択手段が選択したタスクの、命令指定手段が指定する命令を実行する実行手段とを備え、
前記サイクル数は一のタスクが実行され始めてから次のタスクに切り替えられるまでの最大サイクル数であり、
全タスクの最大サイクル数の合計時間は非同期事象の発生周期よりも短く設定されていることを特徴とするプロセッサ。
【請求項２５】　実行対象タスクの一つは前記非同期事象に関する処理が割り当ててあり、そのタスクに割り当てられたサイクル数と非同期事象発生周期内にそのタスクが繰り返し選択される回数との積で与えられるサイクル数が非同期事象に対して所要な処理量を上回っていることを特徴とする請求項２４記載のプロセッサ。
【請求項２６】　前記プロセッサは、動画ストリーム及びオーディオストリームを復号するデコード部と、デコード部が復号処理に用いるローカルバッファと、外部メモリとを有するAVデコードシステムに備えられ、
前記タスク数nは4であり、
n個のタスクにおける第１タスクは、AVデコードシステム外部から入力されてくるビットストリームから動画ストリーム及びオーディオストリームを抽出して、抽出された動画ストリーム及びオーディオストリームを外部メモリに書き込むタスクであり、
第２タスクは、AVデコードシステムにおけるデコード部の進捗に応じて、次に処理すべき動画ストリーム及びオーディオストリームを外部メモリからデコード部に供給するタスクであって、デコード結果を外部メモリに格納するタスクであり、
第３タスクは、
デコード部のデコード結果のうち動画データを外部メモリから読み出して、外部に接続されたディスプレィの同期信号に同期させつつ出力するタスクであり、第４タスクは、
デコード部のデコード結果のうち音声データを外部メモ

リから読み出して、外部に接続されたスピーカー装置に所定の周期で出力するタスクであり、
前記サイクル数保持手段は、
第１～第４タスクに割り当てられたサイクル数として4サイクルを保持することを特徴とする請求項２５記載のプロセッサ。

【請求項２７】　n個のタスクを実行対象タスクとしたプロセッサであって決められたサイクル数でタスクを順次選択するタスク選択手段と、
タスクと一対一の関係でｎ個の命令指定情報を有し、タスク選択手段によってタスクが選択されるとそれに対応した命令指定情報を有効にすると共に、その情報を起点として、次に読み出すべき命令を指定する情報を動的に生成する命令指定手段と、
命令指定手段にて指定された命令を読み出し実行する実行手段とを備えることを特徴とするプロセッサ。

【請求項２８】　前記命令指定手段は、
n個のタスクと一対一に対応し、対応するタスクに関して次に読み出すべきアドレス値を命令指定情報として保持しているｎ個とのアドレスレジスタと、
タスク選択手段が選択したタスクに対応するアドレスレジスタを選択し、そのアドレス値を出力させるレジスタ選択部と、
レジスタ選択部によりアドレスレジスタが選択されると、当該アドレスレジスタが保持しているアドレス値をカウント初期値として保持するカウント値レジスタと、
カウント値レジスタが保持しているカウント値を、サイクル毎に、インクリメントするインクリメンタと、
インクリメントされたカウント値を前記次に読み出すべき命令を指定する情報の更新値として保持する読出先アドレス保持部とを備えることを特徴とする請求項２７記載のプロセッサ。

【請求項２９】　前記命令指定手段はインクリメンタがインクリメントしたカウント値をスルー出力して読出先アドレス保持部に保持させると共に、タスク選択手段がタスクを選択すると、当該タスクに対応するアドレスレジスタのアドレス値を選択出力して読出先アドレス保持部に保持させる第１セレクタと、
次順のタスクに切り換わる際に、カウント値レジスタが保持しているカウント値を用いて、切り換え直前までレジスタ選択部で選択されていたアドレスレジスタが格納するアドレス値を書き換える第１書換部とを備えることを特徴とする請求項２８記載のプロセッサ。

【請求項３０】　前記タスク切り換えを指定する旨の自発切換命令が含まれており、
前記実行手段は読出先アドレス保持部が保持しているアドレスにて特定される命令を解読する解読手段を備え、
命令指定手段は、
解読手段による解読結果が自発切換命令であれば、読出先アドレス保持部が保持している命令指定情報を用いて、切り換え直前まで選択部で選択されていたアドレスレジスタの格納するアドレス値を書き換える第２書換部を備え、
前記第１セレクタは、
解読手段による解読結果が自発切換命令であれば、当該次のタスクに対応するアドレスレジスタのアドレス値を選択出力して読出先アドレス保持部に保持させることを特徴とする請求項２９記載のプロセッサ。

【請求項３１】　前記タスクには絶対アドレスにて分岐先を指定した分岐命令が含まれており、
前記命令指定手段は解読手段による解読結果が分岐命令であれば、当該分岐命令に含まれている絶対アドレスを用いて、切り換え直前まで選択部で選択されていたアドレスレジスタの格納するアドレス値を書き換える第３書換部を備えることを特徴とする請求項３０記載のプロセッサ。

【請求項３２】　前記タスクには間接アドレス指定にて分岐先を指定した分岐命令が含まれており、
前記命令指定手段は解読手段による解読結果が間接アドレス指定型の分岐命令であり、実行手段における命令実行によって分岐先アドレスが算出されれば、当該分岐先アドレスを用いて切り換え直前まで選択部で選択されていたアドレスレジスタの格納するアドレス値を書き換える第４書換部を備えることを特徴とする請求項３１記載のプロセッサ。

【請求項３３】　前記実行手段はn×m個の汎用レジスタから構成され、各タスクにm個の汎用レジスタを割り当てている汎用レジスタセットと、
タスク選択手段によりタスクが選択されると、それに対応するm個の汎用レジスタを用いて選択されたタスクに含まれている命令についての演算を実行する演算実行部とを備えることを特徴とする請求項３２記載のプロセッサ。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】MPEGストリームを始めとする可変符号長のマルチメディアデータを再生するAVデコーダに備えられ、専らAVデコーダ内部のメインプロセッサの周辺制御を担うプロセッサに関する。

【０００２】

【従来の技術】MPEGストリームの再生技術はマルチメディア社会の基盤技術の一つでもあり、近年におけるその需要の高まりには目を見張るものがある。MPEGストリームの再生技術の新たな活躍の舞台として注目を浴びているのは、インタラクティブな動画再生、音声再生を可能とする民生機器の分野であり、この分野にて成功を収めるべく、メーカー各社の技術者はMPEGストリームの再生を高度になしえるAVデコーダの研究開発に心血を注いでいる。

【０００３】MPEGストリームの再生技術にて行うべき処

理には様々なものがあるが、それらはコア処理と、非同期イベント処理とに大別されるといわれる。コア処理とは、逆量子化、逆離散余弦変換、動き補償等、縦16×横16個の画素からなるマクロブロックを対象とした処理である。毎秒4050個ものマクロブロックを処理せねばならないので(4050=30フレーム×30スライス×45マクロブロック)、その演算量は膨大なものとなる。コア処理の実行には、パイプライン処理が好適とされるが、ハードウェア規模の増大を厭わない場合は、解読器や演算器を複数にしてコア処理の負荷を分散させるのが良い。

【０００４】非同期イベント処理とは、複数の要因が重なることにより特定の事象が成立している期間において集中的に行うべき処理、或は、所定の周期をもって間欠的に行うべき処理であり、コア処理と同期して行なえない処理の総称をいう。AVデコーダにおいて、このような非同期イベント処理に相当する処理には、記録媒体や通信媒体からのMPEGストリーム入力に関するもの(1)、AVデコーダから映像再生機器、音声再生機器への出力に関するもの(2)、AVデコーダと、AVデコーダ外部に接続された拡張メモリとの入出力に関するもの(3)の三種類がある。

【０００５】記録媒体からのストリーム入力に関する非同期イベント処理(1)は更に、光ディスク等の記録媒体や通信媒体からMPEGストリームを取り込んで、エレメンタリィストリームを抽出する抽出処理(1-1)、抽出されたエレメンタリィストリームを拡張メモリとして接続されているSDRAMに書き込む書込処理(1-2)がある。再生出力に関する非同期イベント処理(2)は更に、復号が済んだ動画ストリーム、オーディオストリームを映像信号、音声信号にそれぞれ変換してディスプレィ、スピーカに出力する処理(2-1)と、MPEGストリームのプライベートストリームとして出力されてくる副映像を描画し、これを動画データと混合して映像信号に字幕を合成する処理(2-2)を含む。

【０００６】拡張メモリとの入出力に関する非同期イベント処理(3)は更に、逆量子化、逆離散余弦変換、動き補償等の処理が施されたデータが内部バッファに蓄積されるのを監視し、その蓄積量が一定量になった段階でSDRAMにまとめて書き込む処理(3-1)、逆量子化、逆離散余弦変換、動き補償の進捗に応じて、次に処理すべきデータを外部に接続されたSDRAMから内部バッファに間欠的に補充する補充処理(3-2)がある。

【０００７】以上の処理の他にも、操作者からの操作に応じて行うべき処理はこの非同期イベント処理に分類される。単にMPEGストリームの再生だけではなく、操作者の対話性を重視した再生装置の応用システムを設計する場合やホストコンピュータとの同期制御を前提にしたシステムを設計する場合は、更に多くの非同期イベント処理をAVデコーダに処理させる必要がある。

【０００８】上記の処理のうち、処理(2-1)のオーディオに関する処理は、オーディオアウトタスクと呼ばれ、90μsecという間欠的な周期を有する。また処理(2-1)のうち、ビデオに関する処理はビデオアウトタスクと呼ばれ、ディスプレィの水平同期信号の間欠的な周期に基づいた50μsecにおいて、一ライン分の画像の処理を完遂させねばならない。

【０００９】このようにビデオアウトタスク、オーディオアウトタスクに実行周期が与えられるのは、これらのタスクの処理が完遂されないと、動画-音声が時間の流れに沿って円滑に再生されないからである。即ち、ビデオアウトタスク、オーディオアウトタスクの処理を50μsec、90μsecという周期内に完遂することが、動画再生、音声再生のリアルタイム性を満たすための最大要件となる。

【００１０】従来のMPEGストリーム再生技術においては、以上の非同期イベント処理を汎用プロセッサに行わせる場合、非同期事象の発生や所定周期の到来を割込信号により汎用プロセッサに通知する。通知後、汎用プロセッサは分岐命令を用いて非同期イベント処理へと分岐する。

【００１１】

【発明が解決しようとする課題】しかしながら割込信号を用いて汎用プロセッサに非同期イベント処理を実行させるという従来の方式では、オーディオアウトタスク、ビデオアウトタスクの処理を完遂させるには動作クロックをいくらに定めればよいかを示す動作クロック数の最適な下限値が一義的に計算できないという問題点がある。

【００１２】最適な下限値が計算できないため、大まかな見積もりで動作クロック数を決定せざるを得ず、動作クロック数をどうしても高めに設定してしまう傾向がある。従来の方式では、どのような計算法で動作クロックを計算していたかを以下に説明する。周期が異なる非同期イベントタスクとして90μsecの実行周期を有するオーディオアウトタスクと、50μsecの実行周期を有するビデオアウトタスクを実行せねばならないものとする。この場合、オーディオアウトタスクは0μsec、90μsec,180μsec,270μsec,360μsec,450μsecというように90μsec毎にオーディオストリームの復号を完遂せねばならず、ビデオアウトタスクは0μsec、50μsec,100μsec,150μsec,200μsec,250μsec,300μsec,350μsecというように50μsec毎に動画ストリームの復号を完遂せねばならない。従来の汎用プロセッサは、２つの実行周期の到来を割込信号にて通知して、それからビデオアウトタスク、オーディオアウトタスクを処理することになる。

【００１３】割込信号によって実行時期の到来を知ってから、オーディオアウトタスク、ビデオアウトタスクを起動しようとすると、各タスクが処理を完遂せねばならないタイムリミットは、自身を起動するための割込信号

の発生タイミングから、次のタスクを起動するための割込信号の発生タイミングまでに定められる。図５７は、これらの周期が割込信号により汎用プロセッサに通知されて、実行する場合を想定したタイミングチャートである。本図において、オーディオアウトタスクービデオアウトタスクの処理完遂のためのタイムリミットがどのように現れるかを説明する。

【００１４】パルスP1の立ち上がり及びパルスP5の立ち上がりが発生して、ビデオアウトタスク、オーディオアウトタスクを起動する旨が汎用プロセッサに通知されると、次の割込信号であるパルスP2の立ち上がりが通知されるまでの50μsec期間がビデオアウトタスク及びオーディオアウトタスクの処理を完遂させるためのタイムリミットとなる。

【００１５】一方、パルスP6の立ち上がりが発生して、オーディオアウトタスクを起動する旨が汎用プロセッサに通知されると、次の割込信号であるパルスP3の立ち上がりが通知されるまでの僅か10μsecの期間がオーディオアウトタスク完遂のためのタイムリミットとなる。タイムリミットから動作クロックを計算する方式では、10μsecのような極めて短いタイムリミット、即ち、タイムリミットのワーストケースにおいてもオーディオアウトタスク、ビデオアウトタスクを完遂するだけの動作クロック数が要求される。そうすると動作クロックを一段と高いものに決定せさるを得ない。しかし安易に動作クロックを高速にすれば、消費電力の増大も著しくなり、民生機器の用途に適さなくなる。

【００１６】本発明の目的は、所定の周期で完遂せねばならない処理量が予め定められている非同期イベント処理が複数ある場合に、それぞれの非同期イベントタスクにどれだけのサイクル数を割り当てればよいかを示す最適な下限値を一義的に計算することできると共に、それらの最適な下限値から全タスクを動作させるための動作サイクル数の下限値を一義的に計算することができるプロセッサを提供することである。

【００１７】

【課題を解決するための手段】上記目的は、n個のタスクを実行対象タスクとしたプロセッサであって、n個のタスク識別子を出力し、そのタスクに割り当てられた数の命令が実行されると次のタスク識別子を所定の順序で出力する実行タスク指示手段と、出力されるタスク識別子で特定されるタスク中の実行すべき命令を順次指定する命令指定手段と、指定された命令を実行する実行手段とを備えるプロセッサにより達成される。

【００１８】

【発明の実施の形態】

（第１実施形態）I/Oプロセッサについての説明を開始する前に、AVデコーダがどのような内部構成を有しており、その構成において本発明に係るI/Oプロセッサがどのような役割を果たしているかを説明する。図１にAVデコーダの内部構成を示す。

【００１９】AVデコーダは、コア処理を行うデコーダコア部と、複数の非同期イベントタスクを行うI/O部とかなる。デコーダコア部は、setup部１０４、VLD部１０５、IQ/IDCT部１０６、動き補償部１０７を含み、I/O部は、ストリーム入力部１０１、バッファメモリ１０２、ビットストリームFIFO１０３、ビデオ出力部１０８、オーディオ出力部１０９、I/Oプロセッサ１１３、バッファメモリコントローラ１１０、RAMコントローラ１１１、FIFOコントローラ１１２、ホストI/O部１１５を備えている。またバスとしてバッファメモリバス１２１、SDRAMバス１２２、ビットストリームバス１２３、IOPコントロールバス１２４が設けられており、拡張メモリとしてSDRAM３００と接続している。

【００２０】(1.)MPEGストリームの復号処理の概要

次に図２を参照して、MPEGストリームの復号処理の概要について簡単に説明しておく。本実施形態においてMPEGストリームは、動画ストリーム、オーディオストリーム、副映像ストリームからなる。このうち動画ストリームは、圧縮された動画像データであり、復号処理において、差分算出の基準となる画像（以下「参照画像」と呼ぶ）と、算出された差分を符号化した画像（以下「差分画像」と呼ぶ）とを加算することにより、動画像に復元される。

【００２１】図２において、MPEGストリーム内の動画ストリームのデータ構造は、階層的に表現されている。第１層はMPEGストリームの階層であり、第２層は１秒長の動画の階層である。第３層は１フレームの階層であり、第４層は１スライスの階層である。第５層はマクロブロックの階層である。破線C1に示す第１層と第２層との対応関係を参照すると、MPEG(Motion Picture Image Coding Experts Group)において1秒長の動画は、NTSC方式における30フレームの画像（PAL方式では25フレームの画像）にて構成されることがわかる。また、各フレームの画像は、Iピクチャ(図中のIO)、Pピクチャ（図中のP3）、Bピクチャ（図中のB1,B2）という３つのタイプがあることもわかる。

【００２２】ここでＩピクチャとは、圧縮された画像データであるが、一フレーム分の輝度成分及び色差成分を含むものをいう。Pピクチャ（Predictive-Picture ）又はBピクチャ（Bidirectionally predictive Picture）は差分画像と呼ばれるものである。ここでPピクチャとは、過去方向に位置するフレームとの差分からなる差分画像であり、Bピクチャとは、過去方向及び未来方向に位置するフレームとの差分からなる差分画像である。

【００２３】Bピクチャ及びPピクチャがどのような単位で作成されるかであるが、破線C2に示す第２層と第３層との対応関係を参照すると、NTSC方式においてこれらの画像は30本のスライスというデータにて構成され、PAL方式では36本のスライスというデータにて構成されるこ

とがわかる。破線C3に示す第３層と第４層との対応関係を参照すると、各スライスは、４５のマクロブロックにて構成されることがわかる。マクロブロックとは、横16画素×縦16画素の輝度成分、色差成分にて構成される。破線C4に示す第４層と第５層との対応関係を参照すると、横16×縦16の輝度成分からなる輝度ブロックと、横8×縦8の青色差成分からなる青色差ブロック（Cbブロック）と、横8×縦8の赤色差成分からなる赤色差ブロック（Crブロック）とを含むことがわかる。本マクロブロックは、AVデコーダにおける画像復号の一単位となる。

【００２４】マクロブロックより下は、AVデコーダの構成要素間がマクロブロックに含まれている輝度ブロック、色差ブロックをどのように処理するかを示すタイミングチャートになっている。マクロブロックに対してどのような復号化がなされるかであるが、マクロブロックに対する復号化は、圧縮動画像データを可変長復号（Variable Length codeDecoding:以降ＶＬＤと略す）して、６つの空間周波数成分データ、ヘッダ情報、動きベクトルを得る。

【００２５】その後、６つの空間周波数成分データに対して逆離散余弦変換処理(DiscereteCosine Transform:DCT)を行って低周波数帯に位置する空間周波数成分と、高周波数帯に位置する空間周波数成分とに分離し、高周波数帯を切り捨てて、低周波帯に位置する空間周波数成分に対して逆量子化処理を行うことによりなされる。符号化されたマクロブロックは、逆量子化及び逆離散余弦変換処理がなされた後、動きベクトルに基づいた動き補償により、映像として表示される。動きベクトルとは、前後のフレームの画像と比較して相関性の最も高い箇所を指し示す情報をいう。つまり動きベクトルは画像内の人物像や建造物像がフレームの前後でどう動いたかをブロック単位で表現している。

【００２６】動き補償とは、差分の前後に表示されるべき参照画像と、当該差分とを足し合わせて（ブレンドして）一枚の完結した表示用画像を得ることをいう。以上のMPEG準拠の動画像の符号化･複号化技術のうち、本明細書で特に引用する内容は以上の通りである。より詳細内容な技術内容については、株式会社アスキー「ポイント図解式最新ＭＰＥＧ教科書」などの公知文献を参照されたい。

【００２７】図２におけるタイミングチャートは、デコーダコア部におけるsetup部１０４、VLD部１０５、IQ/IDCT部１０６、動き補償部１０７の処理がどのようなタイミングが行われるか説明する際に引用するものとする。以上でMPEGストリームの復号処理の概要について説明を終え、続いてAVデコーダの構成要素についての説明を開始する。

【００２８】(2.)AVデコーダの構成要素

ストリーム入力部１０１は、記録媒体や通信媒体からMPEGストリームが取り出されると、MPEGストリームをAVデコーダ内部に取り込み、I/Oプロセッサ１１３による制御に応じてバッファメモリバス１２１に出力する。バッファメモリ１０２は、バッファメモリコントローラ１１０の制御に従って、ストリーム入力部１０１が取り込んだMPEGストリームを保持する。またMPEGストリームを出力するようバッファメモリコントローラ１１０に命じられると、これまでに格納しているMPEGストリームをSDRAMバス１２２に出力する。

【００２９】ビットストリームFIFO１０３は、MPEGストリームに含まれているエレメンタリィストリームがSDRAMバス１２２に出力されると、FIFOコントローラ１１２による制御に従って出力されたエレメンタリィストリームを取り込む。ビットストリームFIFO１０３は取り込んだエレメンタリィストリームを先入れ先出し方式に保持する。このようにして保持したエレメンタリィストリームをFIFOコントローラ１１２の制御に従って、ビットストリームバス１２３に出力する。

【００３０】setup部１０４は、ビットストリームFIFO１０３に保持されているMPEGストリームからエレメンタリィストリームが抽出されるのを待ち、そのエレメンタリィストリームが動画ストリーム或はオーディオストリームなら、VLD部１０５による復号によりそのヘッダ部が伸長されるのを待つ。ヘッダが伸長されるとこれの解析処理e1を図２に示すように行う。またエレメンタリィストリームが動画ストリームなら動きベクトルの抽出e2を行う。その後、逆量子化、逆離散余弦変換、動き補償等が行われている間、音声ストリームの復号処理e3を行う。

【００３１】VLD部１０５は、ビットストリームFIFO１０３に格納されているエレメンタリィストリームが動画ストリームであり、これを構成するマクロブロックがビットストリームバス１２３に出力されると、マクロブロックに含まれている４つの輝度ブロックY0,Y1,Y2,Y3と、２つの色差ブロックCb,Crとに対して可変符号長デコードt21,t22,t23,t24,t25,t26を行う。

【００３２】IQ/IDCT部１０６は、可変符号長デコードがなされた四つの輝度ブロックと、２つの色差ブロックとに対して逆量子化及び逆離散余弦変換を行う。動き補償部１０７は、IQ/IDCT部１０６により逆量子化及び逆離散余弦変換がなされると、これらの処理がなされた輝度ブロック及び色差ブロックに対応する参照画像(Y0,Y1),(Y2,Y3),(Cb.Cr)をAVデコーダ外部に接続されたSDRAM３００から読み出し、輝度ブロック及び色差ブロックと、参照画像とをブレンドする。そしてそのブレンド結果に対してハーフペル補間を行うことにより動き補償を行う。その後、RAMコントローラ１１１を制御することにより動き補償の結果をSDRAM３００に書き込ませる。

【００３３】ビデオ出力部１０８は、動き補償部１０７により参照画像とのブレンドとハーフペル補間とがなされた一フレームの画像を映像信号に変換して外部に接続

されたテレビ受像器等のディスプレイ装置に出力する。オーディオ出力部１０９は、setup部１０４により復号されたオーディオストリームを音声信号に変換して外部に接続されたスピーカ装置に出力する。

【００３４】バッファメモリコントローラ（図中のBMコントローラ）１１０は、I/Oプロセッサ１１３周辺におけるストリーム入力部１０１、ビデオ出力部１０８、オーディオ出力部１０９間のアクセス調停を行うバッファメモリインターフェィスと、バッファメモリ１０２と、ストリーム入力部１０１、ビデオ出力部１０８、オーディオ出力部１０９間のDMA転送機能を有するDMAコントローラとからなる。

【００３５】RAMコントローラ１１１は、SDRAM３００に対してのバーストリードと、バーストライトとが可能なSDRAMインターフェィスと、SDRAM３００－バッファメモリ１０２間、動き補償部１０７－バッファメモリ１０２間のDMA転送機能を有するDMAコントローラとからなる。FIFOコントローラ１１２は、ディアルポートRAMと、当該RAMのRead/Writeを制御するコントローラと、ビットストリームFIFO１０３におけるアクセスアドレスを示すポインタを管理するポインタ管理機能とを有する。

【００３６】I/Oプロセッサ１１３は、AVデコーダにおける6本もの非同期イベントタスクを６本のスレッドに割り当てることにより、非同期イベントタスクを時分割多重にて実行する。各非同期イベントタスクに割り当てたスレッドは、I/Oプロセッサ内部の全ての構成要素が同期制御のために用いているクロック信号のサイクル数にて表現され、その長さは原則4サイクルであるものとする。

【００３７】（3.）I/Oプロセッサに実行が課された非同期イベントタスクとは、どのような処理内容であるかAVデコーダにおいてI/Oプロセッサに実行が課された非同期イベントタスクが、どのような処理内容であるかを逐一説明する。

・ホストI/Oタスク

本タスクは、ホストコンピュータとの通信と、操作者との対話に応じた二次元グラフィックスの描画に関するタスクである。具体的には、ホストI/O部１１５を介したホストコンピュータとの通信処理と、操作者による操作に応じて二次元グラフィックスを描画し、ビデオ出力部１０８に出力する処理とからなる。

・パージングタスク

本タスクはバッファメモリ１０２に入力されたMPEGストリームのパージング処理と、MPEGストリームから動画ストリーム、オーディオストリーム、副映像ストリームといったエレメンタリィストリームの抽出を行う抽出処理に関する。

【００３８】パージングタスクの実行時において、I/Oプロセッサ１１３は外部からストリーム入力部１０１に入力されたMPEGストリームをバッファメモリバス（図中ではBMバス）１２１に出力させる。バッファメモリコントローラ１１０を制御することによりバッファメモリバス１２１に出力されたMPEGストリームをバッファメモリ１０２に書き込ませる。その後I/Oプロセッサ１１３は、バッファメモリコントローラ１１０を制御することによりMPEGストリームをバッファメモリバス１２１に出力させ、MPEGストリームからエレメンタリィストリームを抽出する。バッファメモリコントローラ１１０を制御することにより抽出されたエレメンタリィストリームをバッファメモリ１０２に書き込ませる。その結果、バッファメモリ１０２には、エレメンタリィストリームとしてビットストリーム、オーディオストリーム、副映像ストリームが格納された状態となる。

・オーディオストリーム転送制御タスク

オーディオストリーム転送制御タスクはオーディオストリームに関する全ての転送制御からなるタスクである。

【００３９】オーディオストリーム転送制御タスクに従って、I/Oプロセッサ１１３はバッファメモリコントローラ１１０を制御することによりバッファメモリ１０２にエレメンタリィストリームとして格納されているオーディオストリームをSDRAMバス１２２に出力させる。その後、RAMコントローラ１１１を制御することによりSDRAMバス１２２に出力されたオーディオストリームをSDRAM３００に書き込ませる。setup部１０４がビットストリームFIFO１０３に格納されているオーディオストリームの復号を開始すると、その復号処理がどれだけ進捗したかを監視し、オーディオストリームの残量が所定値以下となると、RAMコントローラ１１１を制御して、SDRAM３００に格納されているオーディオストリームを読み出させてSDRAMバス１２２に出力させる。FIFOコントローラ１１２を制御してこのように出力されたオーディオストリームをビットストリームFIFO１０３に書き込ませる。

【００４０】このようなビットストリームFIFO１０３への書き込みにより、オーディオストリームの復号の進捗に応じたオーディオストリームの補充を行う。デコードが済むと、I/Oプロセッサ１１３はFIFOコントローラ１１２を制御することによりデコード済みオーディオストリームをSDRAMバス１２２に出力させる。バッファメモリコントローラ１１０を制御して、SDRAMバス１２２に出力されたオーディオストリームをバッファメモリ１０２に書き込ませる。

・動画ストリーム転送制御タスク

動画ストリーム転送制御タスクは動画ストリームに関する全ての転送制御からなるタスクである。

【００４１】I/Oプロセッサ１１３は、バッファメモリコントローラ１１０を制御することによりバッファメモリ１０２に格納されている動画ストリームをSDRAMバス１２２に出力させ、RAMコントローラ１１１を制御することによりSDRAMバス１２２に出力された動画ストリームをSDRAM３００に書き込ませる。VLD部１０５、IQ/IDC

T部１０６、動き補償部１０７がビットストリームFIFO１０３に格納されている動画ストリームの復号を開始すると、動画ストリームの復号がどれだけ進捗したかを監視し、復号すべき残量が所定値以下となると、RAMコントローラ１１１を制御して、SDRAM３００に格納されている動画ストリームを読み出させて、FIFOコントローラ１１２にビットストリームFIFO１０３に書き込ませる。このようなビットストリームFIFO１０３の書き込みにより、復号処理の進捗に応じた動画ストリームの補充を行う。

・ビデオアウトタスク

ビデオアウトタスクは映像出力に関する出力制御からなるタスクである。

【００４２】本非同期イベントタスクの実行時においてI/Oプロセッサ１１３はRAMコントローラ１１１を制御することにより、VLD部１０５、IQ/IDCT部１０６、動き補償部１０７による処理が済んでSDRAM３００に格納されている動画データをSDRAMバス１２２に出力させ、バッファメモリコントローラ１１０を制御して、SDRAMバス１２２に出力された動画ストリームをバッファメモリ１０２に書き込ませる。バッファメモリコントローラ１１０を制御して、SDRAM３００に書き込まれた動画ストリームをSDRAMバス１２２に出力させ、ビデオ出力部１０８に映像信号に変換させる。それと共に副映像ストリームを展開して得られた副映像と、二次元グラフィックスとを混合してビデオ出力部１０８に出力させる。

・setupI/Oタスク

setup部１０４とI/Oプロセッサ１１３とが通信する際に実行すべき処理からなる。

【００４３】当該コマンドがホストコンピュータが備えているレジスタを読み書きするためのコマンドなら、I/Oプロセッサ１１３を制御して当該読み書きのための処理と、当該コマンドがVLD部１０５、IQ/IDCT部１０６の初期化コマンドなら、VLD部１０５、IQ/IDCT部１０６をリセットする処理とからなる。以上の非同期イベントタスクは、個々の処理負荷は軽いが、間欠的な起動が要求される。特にビデオアウトタスクは水平ブランキング期間の50μsec毎の起動が要求され、オーディオアウトタスクは90μsec毎の起動が要求される。

【００４４】（4.）I/Oプロセッサがどのように構成されているか

I/Oプロセッサの第１実施形態を図面を参照しながら説明する。上述したように、I/Oプロセッサはそれぞれの非同期イベントタスクを実行すべき事象が成立したか否かの監視を行っており、事象が成立した非同期イベントタスクをより優先的に実行する。本実施形態において優先実行の手順までも言及すると、説明が煩雑になるのでこの優先実行についての説明は第３実施形態において行うものとする。第１実施形態では、I/Oプロセッサの基本構成、即ち、AVデコーダにおける6本もの非同期イベントタスクを時分割多重にて実行する構成について説明する。

【００４５】図３はI/Oプロセッサの内部構成を示す図であり、本図に示すようにI/Oプロセッサは命令メモリ１００、命令読出回路１０、命令解読制御部１１、レジスタセット１２、演算実行部１４、及びタスク管理部１５から構成される。命令メモリ１００は、上述した六本の非同期イベントタスクを構成する命令を記憶する。図８は、命令メモリ１００がどのように命令列を記憶しているかを示す図である。本図において各命令には、0-0,0-1,0-2といった識別子が付されている。この識別子のうち、上位の数は、当該命令が６本の非同期イベントタスクの何れに属するかを示し、下位の数は、属する非同期イベントタスクにおいて当該命令が何番目に位置するかを示す。以降の説明において、命令メモリ１００における各命令の配置アドレスは、この識別子を用いて表すものとする。

【００４６】命令メモリ１００内に記憶されている命令は、全てのオペレーションが一命令語長に収まるよう命令フォーマットが規定されている。本実施形態において命令語長は16bitに規定されており、その命令フォーマットを図９～図１１に示す。図９は、レジスタをオペランドに指定した算術演算命令の命令フォーマットを示す。本命令フォーマットでは、2bit～4bitに第1オペランドに指定すべきレジスタ番号を指定でき、5bit～7bitには、第2オペランドに指定すべきレジスタ番号を指定できる。8bit～10bitには、演算結果の格納先に指定すべきレジスタ番号を指定でき、11bitから15bitまでの5bitにオペレーション内容が指定される。下段の表は、当該オペレーション内容の指定が可能な5bitのうち、上位2bitと、下位3bitの組み合わせと、その組み合わせにて表現されるオペレーション内容とを示している。例えば上位2bitが『00』であり、下位3bitが『000』である場合、その組み合わせは加算演算のオペレーションを表現しており、上位2bitが『01』であり、下位3bitが『000』である場合、その組み合わせは減算演算を表現している。上位2bitが『00』であり、下位3bitが『001』である場合、その組み合わせは左シフト演算（図中の<<は左シフト演算の意味である）を表現しており、上位2bitが『01』であり、下位3bitが『001』である場合、その組み合わせは右シフト演算（図中の>>は右シフト演算の意味である）を表現している。

【００４７】図１０は、レジスタにて読出先アドレス、書込先アドレスを指定したメモリロード命令／メモリストア命令の命令フォーマットを示す。この命令フォーマットでは、2bit～4bitに第1オペランドに指定すべきレジスタ番号を指定でき、5bit～7bitに読出元に指定すべきレジスタ番号又は演算結果の格納先に指定すべきレジスタ番号を指定できる。10bitから12bitまでの3bitにオペレーション内容が指定され、13bitから15bitまでの3b

itに3bit長の即値が指定される。

【００４８】下段の表は、3bitによるオペレーション内容の指定のうち、上位1bitと、下位2bitの組み合わせと、その組み合わせにて表現されるオペレーション内容とを示している。例えば上位1bitが『1』であり、下位2bitが『10』である場合、読出元オペランドsrc_REGで指定されたレジスタの値を、第1オペランドsrc1と3bitの即値im:3との組み合わせ(src1+im:3)を用いて指定されたメモリに格納するストア命令を指定している。

【００４９】上位1bitが『0』であり、下位2bitが『10』である場合、その組み合わせは第1オペランドsrc1と3bitの即値との組み合わせ(src1+im:3)を用いたアドレッシングモードによりメモリから値を読み出し、格納先dst_REGに指定されたレジスタに格納するロード命令を指定している。図１１は、8bit、11bitの即値の指定が可能な比較命令／演算命令／分岐命令の命令フォーマットを示す。本命令フォーマットにおいて即値が8bitである場合は、5bitから7bitまでに第１オペランド及び演算結果の格納に相当するレジスタの指定が可能であり、8bitから15bitまでを即値の格納に用いる。即値が11bitである場合は、5bitから15bitまでを即値の格納に用いる。2bitから5bitまでの4bitを用いて下段の表のようにオペレーション内容が指定される。

【００５０】下段の表は、当該4bitのうち、上位2bitと、下位2bitの組み合わせと、その組み合わせにて表現されるオペレーション内容とを示している。例えば上位2bitが『10』であり、下位2bitが『01』である場合、その組み合わせは第１オペランドで指定されたレジスタの値を第２オペランドと8bitの即値とを比較する比較命令を指定している。

【００５１】上位2bitが『10』であり、下位2bitが『10』である場合、その組み合わせはオペランドと8bitの即値とを乗算する乗算命令を指定している。上位2bitが『11』であり、下位2bitが『11』である場合、その組み合わせは11bitの即値を分岐先アドレスとした絶対アドレス指定型の分岐命令を指定している。以上の説明において、オペレーション内容は一命令語長である16bitにて表現されるため、命令メモリ１００内の如何なるオペレーション内容の命令を読み出す場合でも、またどの命令を解読･実行する場合でも、それらの読出、解読、実行が均一に一サイクル内に完遂することができる。即ち、命令の読出･解読･実行が完遂するまでの時間が命令間でバラツキが生じないように命令フォーマットが規定されているのである。特に図１１において即値の幅を8bit、11bitとしていることに、この規定の意図は大きく現れている。つまり指定される即値を長くすると、命令語長が24bit、32bitまで拡張されてしまう恐れがあり、その拡張を防ぐため、命令語長が16bitに収まるように、即値長を8bit、11bitに留めているのである。一つの命令の実行に、どれだけのサイクル数が必要であるかを示す単位(CPI(Cycle Per Instruction)という)にて表現すると、本I/Oプロセッサの CPIは純粋に『1』であり、『命令を何回読み出したか』『命令を何回解読･実行したか』はサイクル数にて表現される。

【００５２】命令読出回路１０は、命令メモリ１００内の読み出し先アドレスを命令メモリ１００に出力する。命令解読制御部１１（図中ではDecoder１１）は、命令メモリ１００が出力する命令を解読し、その解読結果に従って命令読出回路１０及び演算実行部１４を制御する。

【００５３】レジスタセット１２は、32bit長の汎用レジスタ(GR)と、16bit長の汎用レジスタとをそれぞれ24本有する。このように24本ものレジスタを有しているのは、32bit長のレジスタと、16bit長のレジスタとを４本ずつ、６本の非同期イベントタスクに割り当てるためである。このように各非同期イベントタスク毎に８本のレジスタを割り当てることにより、タスクの切り換え時にレジスタの値を退避･復元する必要がなくなる。

【００５４】演算実行部１４は、ALU,乗算器,バレルシフタを備え、命令解読制御部１１による制御に基づきレジスタセット１２における汎用レジスタの格納値を用いて演算を行う。非同期イベントタスクから別タスクへの切り換えられた際、演算実行部１４はそれに対応する４本の32bit長の汎用レジスタ、４本の16bit長の汎用レジスタを用いて、切り換え後のタスクに含まれている命令についての演算を実行するタスク管理部１５は、命令解読制御部１１における命令実行を監視し、各タスクに割り当てられたスレッドの時間長が経過すれば次にスレッドを割り当てるべきタスクの識別子を命令読出回路１０に出力する。

(4.1) 命令読出回路１０は、読出先アドレスの更新をどのように行うか

命令読出回路１０がどのような内部構成にて読出先アドレスの更新を行うかを図４を参照しながら説明する。命令読出回路１０の内部構成を図４に示す。

【００５５】図４に示すように、命令読出回路１０はIF1+1保持部２０、increment回路２１、IF2保持部２２、DECPC保持部２３、タスク別PC格納部２４、セレクタ２５及びセレクタ２６からなり、解読ステージの前に二段の読出ステージを行うパイプライン処理を実現するよう構成されている。この二段の読出ステージにおけるそれぞれの読出先アドレスを第１読出先アドレスIF1、第２読出先アドレスIF2という。

【００５６】図４においてIF1はセレクタ２６が出力したアドレスをいい、IF2はIF2保持部２２が保持するアドレスをいう。図１２は、命令読出回路１０内で行われるパイプライン処理を示すタイミングチャートである。以降、命令読出回路１０の構成要素について言及する場合は、本タイミングチャートを引用するものとする。IF1+1保持部２０は、カウント値を保持するためのカウント

値レジスタとして用いられるものであり、タスク別PC格納部２４から何れか一つの読出先アドレスが出力されると、当該アドレスをカウント初期値として保持し、そのカウント値のインクリメントがincrement回路２１より行われると、インクリメント後のアドレスを最新のカウント値として保持する。

【００５７】この際、increment回路２１より新たにインクリメントされたアドレスが出力されてくると、IF1+1保持部２０はそれまで保持していたアドレスを信号線④を介してセレクタ２５に出力すると共にセレクタ２６に出力する。increment回路２１は、セレクタ２６により出力されたIF1をクロック信号に従ってインクリメントする。インクリメントされたアドレスはIF1+1保持部２０により保持される。セレクタ２６により出力されたIF1は、それまでIF1+1保持部２０により保持されていた値であるので、increment回路２１によるインクリメントにより、IF1+1保持部２０が保持していたカウント値は一クロック毎にインクリメントされてゆく。

【００５８】図１２のタイミングチャートにおいて、読出先アドレスとして出力された命令0-0の読出先アドレスは、increment回路２１によってインクリメントされる（図中のinc1,inc2,inc3参照）。これによりIF1+1保持部２０が保持する読出先アドレスは、命令0-1のアドレス、命令0-2のアドレス、命令0-3のアドレスというように更新されてゆく。

【００５９】IF2保持部２２は、クロック信号に同期して前回IF1+1保持部２０により出力されたアドレスを命令を読み出すべき第２の読出先アドレスとして保持する。図１２のタイミングチャートでは、第２の読出先アドレスとして命令0-0のアドレス、命令0-1のアドレス、命令0-2のアドレス、命令0-3のアドレスが出力されていることがわかる。これらのアドレスは第１の読出先アドレスより一サイクルだけ遅れていることがわかる。新たなアドレスがIF1+1保持部２０から出力されると、IF2保持部２２はそれまで保持していたアドレスをDECPC保持部２３に出力し、信号線①を介してセレクタ２５にも出力する。出力後、IF1+1保持部２０により出力されたその新たなアドレスを保持する。

【００６０】DECPC保持部２３は、前回IF2保持部２２により出力されたアドレスを保持する。このアドレスは命令解読制御部１１による解読制御の対象となる命令のアドレスと一致する。新たなアドレスがIF2保持部２２により出力されると、その新たなアドレスを保持して解読対象のアドレスの更新を行う。DECPC保持部２３が保持するアドレスは、IF2保持部２２がそれまで保持していたアドレスであり、IF2保持部２２が保持するアドレスは、IF1+1保持部２０がそれまで保持していたアドレスであるから、IF1+1保持部２０が保持しているアドレスと比較すると、DECPC保持部２３に保持されているアドレスは二命令遅れであり、IF2保持部２２が保持しているアドレスと比較すると、DECPC保持部２３に保持されているアドレスは一命令遅れである。

【００６１】タスク別PC格納部２４は、タスク毎の読出先アドレスを保持する領域をその内部に有するメモリ回路或はタスク毎に設けられたアドレスレジスタである。内部領域或は個々のアドレスレジスタにおいて、個々に記憶されている読出先アドレスには、3ビット長のタスク識別子がアドレスとして付されている。尚、タスク(0)の読出先アドレスをIPC0と呼び、タスク(1)の読出先アドレスをIPC1と呼ぶ。以降タスク(2)、タスク(3)、タスク(4)、タスク(5)についても同様である。

【００６２】タスク別PC格納部２４におけるこれらの読出先アドレスの読み書き動作はタスク管理部１５からタスクを切り換える旨の指示（タスク切替信号chg_task_exのHigh値）が出力された場合に行われる。このタスク切替信号chg_task_exがHigh値である場合においてタスク管理部１５よりタスク切替信号chg_task_exと共に実行が済んだタスクの3ビット長の識別子（書き込みアドレス選択信号taskid(wr_adr)）及び次に実行すべきタスクの3ビット長の識別子（読み出しアドレス選択信号nxttaskid(rd_adr)）が出力されるが、タスク別PC格納部２４はこれらの信号に従った内部領域の読み書きを行う。即ちタスク別PC格納部２４は書き込みアドレス選択信号taskid(wr_adr)をタスク別PC格納部２４内のアドレスとして解釈して、それに指示される内部領域にセレクタ２５から出力された読出先アドレスを格納する。図１２のタイミングチャートにおいてタスク切替信号chg_task_exが立ち上がったタイミングa1,b1において読出先アドレスの書き込みa2,b2が行われる。

【００６３】書き込みa2が行われた時期には、IF2保持部２２は命令0-3のアドレスを保持しており、IF1+1保持部２０はこの命令0-3のアドレスに『1』を加算したアドレス0-4を信号線④を介してセレクタ２５に出力している。このように出力されると、アドレス0-4がタスク別PC格納部２４に格納される。これにより、タスク別PC格納部２４内のタスク(0)に次のスレッドが巡ってくれば、タスク(0)の読出はアドレス0-4から行われる。

【００６４】書き込みb2が行われた時期には、IF2保持部２２は命令1-3のアドレスを保持しており、IF1+1保持部２０はこの命令1-3のアドレスに『1』を加算したアドレス1-4を信号線④を介してセレクタ２５に出力している。このように出力すると、インクリメントされたアドレス1-4がタスク別PC格納部２４に格納される。これにより、タスク別PC格納部２４内のタスク(1)に次のスレッドが巡ってくれば、タスク(1)の読出はアドレス1-4から行われる。

【００６５】また読み出しアドレス選択信号nxttaskid(rd_adr)がタスク管理部１５より与えられると、タスク別PC格納部２４は、その識別子をタスク別PC格納部２４内のアドレスとして解釈して、それにより指示される内

部領域から非同期イベントタスクの読出先アドレスを取り出してセレクタ２６に出力する。図１２のタイミングチャートにおいて、読み出しアドレス選択信号nxttaskidとして識別子(1)がタスク管理部１５より与えられ、chg_task_exが立ち上がると（参照符号a3参照）、タスク別PC格納部２４は、その識別子により指示されるタスク(1)をタスク別PC格納部２４内のアドレスとして解釈して、そこに格納されている読出先アドレス(1-0)を取り出してセレクタ２６に出力する（参照符号a4参照）。

【００６６】命令2-0のアドレスが読出先アドレスIF1として出力されたタイミングで、読み出しアドレス選択信号nxttaskidとして識別子(2)がタスク管理部１５より与えられ、chg_task_exが立ち上がると（参照符号a5参照）、タスク別PC格納部２４は、その識別子により指示されるタスク(2)の読出先アドレスとして読出先アドレス(2-0)を取り出してセレクタ２６に出力する（参照符号a6参照）。

【００６７】以上のように読出先アドレスが出力されると、タスク(0)の命令は４命令だけ実行され、これに続いてタスク(1)の命令が４命令だけ実行される。以降、タスク(2)、タスク(3)、タスク(4)、タスク(5)の命令列がそれぞれ四命令ずつ実行されてゆく。4入力－1出力のセレクタであるセレクタ２５は、①、②、③、④の信号線に伝送されるアドレスの何れかを選択的にタスク別PC格納部２４に出力する。これらの①～④の何れを選択するかは図５に示す出力論理表に示す通りであり、命令解読制御部１１が命令の解読結果に応じて出力するselpc信号に基づいてなされる。ここで④の信号線はIF1+1保持部２０の出力であり、タスク管理部１５よりタスク切替信号chg_task_exがHigh値に切り換えられた場合はセレクタ２５はこれを選択する。

【００６８】また②、③の信号線はそれぞれ命令解読制御部１１及び演算実行部１４を接続されている。②は、絶対アドレス指定を用いた分岐命令の解読時において、分岐命令の即値を分岐先アドレスとして命令解読制御部１１から取得するために選択される。③は、間接参照指定を用いた分岐命令の解読時において、演算実行部１４により算出されたアドレスを分岐先アドレスとして用いる際に選択される。

【００６９】セレクタ２６は、2入力－1出力のセレクタであり、タスク切替信号chg_task_exがLowの期間においてincrement回路２１によるインクリメント後のアドレスを選択してIF1としてincrement回路２１及びIF2保持部２２に出力する。タスク切替信号chg_task_exがHighの期間においてタスク別PC格納部２４に格納されている次に実行すべきタスクのアドレスをIF1としてincrement回路２１及びIF2保持部２２に出力する。

【００７０】以上のように構成された命令読み出し回路によれば、タスク切り換え時において、これまで実行されていたタスクの読出先アドレスはセレクタ２５によりタスク別PC格納部２４に格納され、これに代えてタスク別PC格納部２４に格納されている次のタスクのアドレスがセレクタ２６によりIF1として出力される。以上の読出先アドレスの切り換えは、一サイクルにて行われ、この読出先アドレスの切り換えにおいて、I/Oプロセッサは余計な処理を行う必要がない。割り込み信号の通知時に行われる分岐のように先行的に読み出したアドレスを無効化することもない。従って、命令読出回路１０による読出先アドレスの退避･復元は、オーバーヘッドの発生無しに行われる。

(4.2) タスク管理部１５は、次に実行すべき非同期イベントタスクをどのように決定するか

タスク管理部１５がどのような内部構成にて次に実行すべき非同期イベントタスクを決定するかを図６に示すタスク管理部１５の内部構成を参照しながら説明する。

【００７１】図６は、タスク管理部１５の内部構成を示す図である。図６に示すようにタスク管理部１５はスレッドマネージャー６１及びスケジューラ６２からなり、スレッドマネージャー６１はフリップフロップ ５１、カウンタ５２、及び比較器５４から構成され、スケジューラ６２はタスクID保持部７１、タスクID保持部７２、タスクラウンド管理部７３、及びプライオリティエンコーダ７４から構成される。

【００７２】フリップフロップ ５１は、次に実行される命令が何番目であるかを示す整数値（これをカウント値iと呼ぶ）を保持する。カウンタ５２は、初期値が『1』に設定され、上限値が『4』に設定されたカウンタであり、クロック信号に同期してフリップフロップ ５１に保持されているカウント値を1,2,3,4,1,2,3,4というようにカウントアップする。

【００７３】比較器５４は、カウンタ５２がカウントアップした値と、整数値『４』との比較を行い、もしカウンタ５２のカウント値と整数値『４』とが一致すれば、タスク切替信号chg_task_exをHigh値にしてセレクタ２６及びタスク別PC格納部２４に出力する。このようにカウンタ５２が『４』をカウントする度にタスク切替信号chg_task_exがHighになることにより、セレクタ２６によるタスク別PC格納部２４の選択は、４サイクルにつき一回行われることになる。この際、タスク別PC格納部２４は次にスレッドを割り当てるべきタスクの読出先アドレスをセレクタ２６に出力しているから、タスク別PC格納部２４からの読出先アドレスの出力は、４サイクルの実行につき一回行われることになる。

【００７４】タスクID保持部７１は、命令実行のカウントがカウンタ５２により行われているタスクの識別子（これをタスク識別子taskidと呼ぶ。）を保持する。タスク識別子の一例を図７（ａ）に示す。図７（ａ）に示すように、タスク識別子は3ビットを用いて、命令メモリ１００に記憶されている６つのタスクの何れか一つを指示する。命令実行のカウントが４回行われると、スケ

ジューラ６２においてプライオリティエンコーダ７４から次にスレッドを割り当てるべきタスクのタスク識別子taskidが出力されてくるが、この際タスクID保持部７１は、これまで保持していたタスク識別子taskidをタスクラウンド管理部７３に出力し、新たに出力されてくるタスク識別子taskidを保持する。

【００７５】タスクラウンド管理部７３は、タスクID保持部７１においてタスク識別子がどのように更新されてきたかを監視し、6ビット長のレジスタを用いて六個のタスクのうちスレッドが割り当てられたものを管理する。またカウンタ５２が4命令の実行をカウントすると、タスクID保持部７１によりタスク識別子taskidが出力され、タスクラウンド管理部７３はスレッドの割り当てが済んだタスクを示す数値（この数値をラウンド値taskn と呼ぶ）をプライオリティエンコーダ７４に出力する。ラウンド値taskn の一例を図７（ｂ）に示す。このラウンド値taskn において、第１ビットが『1』であれば、タスク(1)にスレッドが割り当てられたことを示し、第１ビットが『0』であればタスク(1)がそうでないことを示す。タスク管理レジスタ１３の第２ビットが『1』であれば、タスク(2)にスレッドが割り当てられたことを示し、第２ビットが『0』であればタスク(2)がそうでないことを示す。

【００７６】タスク(0)、タスク(1)にスレッドの割り当てが済めば、タスクラウンド管理部７３は『000011』をプライオリティエンコーダ７４に出力する。またタスク(0)、タスク(1)、タスク(2)にスレッドの割り当てが済めば、タスクラウンド管理部７３は『000111』をプライオリティエンコーダ７４に出力し、タスク(0)、タスク(1)、タスク(2)、タスク(3)にスレッドの割り当てが済めば、タスクラウンド管理部７３は『001111』をプライオリティエンコーダ７４に出力する。

【００７７】尚タスク(0)～タスク(5)へのスレッドの割り当てが一巡すると、『111111』をプライオリティエンコーダ７４に出力した後6ビット長のレジスタを『000000』にリセットする。プライオリティエンコーダ７４は、タスクラウンド管理部７３が出力したラウンド値taskn を受け取り、受け取ったタスクラウンド管理部７３のラウンド値taskn において何ビット目に『1』から『0』への反転が生じているかを検出する。このように『0』への反転が生じたビットを検出すると、この反転が生じたビットに割り当られているタスクのタスク識別子taskidをタスクID保持部７１に出力する。出力されたタスク識別子は、アドレス選択信号nxttaskid(rd_adr)としてタスク別PC格納部２４に出力される。

【００７８】例えばタスクラウンド管理部７３からラウンド値taskn として『000011』が出力されると、プライオリティエンコーダ７４はこのラウンド値taskn において下位から第２ビット目に『0』への反転が生じていることを検出する（尚、最下位ビットをゼロビットと数えていることは留意されたい。）。この第２ビット目は図７（ｂ）においてタスク(2)に割り当られているビットであるから、プライオリティエンコーダ７４はタスク(2)のタスク識別子taskidをタスクID保持部７１に出力する。

【００７９】例えばタスクラウンド管理部７３からラウンド値taskn として『000111』が出力されると、プライオリティエンコーダ７４はこのラウンド値taskn において下位から第３ビット目に『0』への反転が生じていることを検出する。この第３ビット目は図７（ｂ）においてタスク(3)に割り当られているビットであるから、プライオリティエンコーダ７４はタスク(3)のタスク識別子taskidをタスクID保持部７１に出力する。ラウンド値taskn は既にスレッドの割り当てが済んだタスクを示すから、このラウンド値taskn において『0』へと反転しているビットは次にスレッドを割り当てるべきタスクがどれであるかを意味する。『0』へと反転しているビットをタスク識別子taskidに変換すると、タスクID保持部７１には次にスレッドを割り当てるべきタスクが格納される。

【００８０】タスクID保持部７２は、プライオリティエンコーダ７４が新たなタスク識別子taskidを出力すると、直前に出力されていたタスク識別子taskidを書き込みアドレス選択信号taskid(wr_adr)として保持して、タスク別PC格納部２４に出力する。以上のように構成された第１実施形態のI/Oプロセッサによって、どのようにスレッドが割り当てられるかを図１３に示す。図１３は、タスク(0)からタスク(5)までのタスクに割り当てたスレッドを時間順に並べることにより形成されるフレームという単位を示す。

【００８１】タスク(0)～タスク(5)にはそれぞれ4サイクルのスレッドが割り当てられるので、２４サイクルのフレームが形成されることになる。このフレームにより、図７に示した命令列は四命令ずつ均一に実行されることになる。

(5.) 動作クロック数の計算について

第１実施形態のI/Oプロセッサにおいて動作クロック数がどのように計算されるかについて図５５を参照しながら説明する。図５５は第１実施形態におけるスレッド割り当てにてビデオアウトタスク、オーディオアウトタスクを実行する場合に動画ストリーム、オーディオストリームがどのように処理されるかを示すタイミングチャートである。ここでビデオアウトタスク、オーディオアウトタスクのリアルタイム性を保証するには、動画ストリームを映像信号に変換するためのビデオアウトタスクはディスプレィの水平同期信号の周期に基づいた50μsecにおいて、一ライン分の画像を処理させねばならないとし、オーディオストリームを音声信号に変換するためのオーディオアウトタスクは90μsecという周期において、所定の音声ストリームを処理せねばならないとす

る。

【００８２】この場合、先ず第１に上記の周期において『これだけの数の命令は最低実行しておかねばならない』という最低消化数を導出する。ここで一ライン分の映像データを復号するには、ビデオアウトタスク内の448個もの命令実行を最低消化せねばならず、一周期分の音声データを復号するには、オーディオアウトタスク内の808個もの命令実行を最低消化せねばならない。また、本実施形態においてCPI＝１であり、命令数はサイクル数と等価であるため、最低消化数からビデオアウトタスク、オーディオアウトタスクに割り当てるべきサイクル数が448(=4×112)サイクル、808(=4×202)サイクルと決定される。

【００８３】またタスク(0)～タスク(5)はそれぞれを四サイクルずつ実行される。ここで命令読出回路１０内でのセレクタ２５、セレクタ２６によるプログラムカウンタの切り換えによりオーバーヘッドは発生しないので、タスク(0)～タスク(5)の一巡に要する時間は実質24サイクルとなる。命令サイクル数は4であり、タスク総数は6であるので、全てのタスクは、24サイクルにて4命令ずつ実行されることになる。

【００８４】そうすると、1サイクルに要する時間S1,S2は以下の式を満たさねばならない。

$50\mu sec > 6\times4\times112\times S1nsec$

$90\mu sec > 6\times4\times202\times S2nsec$

$S1 < 50\mu sec / 6\times4\times112 = 18.6nsec$

$S2 < 90\mu sec / 6\times4\times202 = 18.56nsec$

動作クロック数を1÷S2nsecの計算から定めると、

54MHz = 1÷18.56nsec

となる。以上の計算により、ビデオアウトタスク、オーディオアウトタスクの処理完遂に最適な動作クロック数を一義的に導出することができる。以上のように本実施形態によれば、動作クロック数の最適な下限値を所定の周期において実行せねばならない命令の最低消化数と、実行周期と、タスク総数とから一義的に導出することができる。動作クロックの最適な下限値が求まるので、動作クロック信号を一段と高いものに定めずとも、オーディオアウトタスク、ビデオアウトタスクのリアルタイム性を保証でき、動作クロックが低速なタイプが多い民生機器に搭載される場合でも、MPEGストリームを好適に復号することができる。

【００８５】また、複数の非同期イベントタスクを時分割多重にて実行するI/Oプロセッサを設けることにより、割込にて非同期イベント処理を処理しなくて済むので、setup部１０４、VLD部１０５、IQ/IDCT部１０６は逆量子化、逆離散余弦変換、動き補償の処理に専念することができる。また、I/Oプロセッサはプログラムカウンタの値を退避し、復元するためのオーバーヘッドの発生無しに高速にタスクを切り換えて実行するので、動作クロックを高速にしなくても、MPEGストリームの再生に要求されるリアルタイム性を満たすことができる。

【００８６】尚、本実施形態において各タスクについてのスレッドを一律に4サイクルとしたが、各タスクの処理内容に応じて、それぞれ異なる値に設定しても良い。タスク別にサイクル数を可変にする場合、図６に示したスレッドマネージャーを図５６に示すように構成する。本図においてセレクタ２００は、タスク(0)からタスク(5)までの固有のサイクル数である『cycle number for task0』～『 cycle number for task5』と接続され、スケジューラ６２が出力したnxttaskidに応じて対応するサイクル数を択一式に比較器５４に出力する。図２６において次にスレッドを割り当てるべきタスクがタスク(0)なら、セレクタ２００は『cyclenumber for task0』を比較器５４に出力する。このように構成すれば、タスク(0)からタスク(5)までのそれぞれの処理負荷に応じた最適なサイクル数を各タスクに分配することができる。またこのような最適なサイクル数をメモリ又はレジスタに記憶させておき、これを後日書き換えてもよい。

【００８７】更に、本実施形態において実行すべきタスクを非同期イベントタスクに絞って説明したが、別のタスクでもよい。

（第２実施形態）第１実施形態では各タスクのスレッドに割り当る命令数を一律四サイクルとしていたが、第２実施形態ではどれだけのサイクルを１スレッドに割り当るかを、各タスクが独自に変更できるようにした実施形態である。スレッドへの割り当ては、Wait_Until_Next命令によりなされる。Wait_Until_Next命令とはスレッドにおける命令を途中で打ち切り、Wait_Until_Next命令の次順に位置する一連の命令列を次のスレッドに先送りする旨の命令である。尚Wait_Until_Next命令の解読時における読み出し先アドレスの格納であるが、この際図４に示した5入力－1出力のセレクタ２５は、①の信号線を介してIF1+1保持部２０の出力を選択してタスクPC格納部２４に格納する。

【００８８】図１４は、Wait_Until_Next命令（図中のWUN命令）を命令0-1としてその内部に有した非同期イベントタスクの一例であり、図１６は、図１４の非同期イベントタスクがどのように実行されるかを示すタイミングチャートである。Wait_Until_Next命令が図１４に示すように命令0-1として非同期イベントタスクの第１番目の命令として格納されている場合、この命令0-1がDEC PC保持部２３に格納され、命令解読制御部１１により解読されると、セレクタ２５は、①の信号線を介して命令0-2の読出先アドレスをタスク別PC格納部２４に格納する（図１６の①も同様である。）。その後命令解読制御部１１は、命令0-2、命令0-3を無効化する（図１６における『NOP』）。

【００８９】ここで命令0-2は、I/Oプロセッサのローカルメモリ内のアドレスmem1の値をレジスタセット１２内のレジスタr0に読み出す命令であり、命令0-3はI/Oプロ

セッサのローカルメモリ内のアドレスbuf1の値をレジスタr1に読み出す命令である。命令0-4はレジスタr0の値とレジスタr1の値とを乗じて、その結果をレジスタr2に格納する命令であり、命令0-5は、レジスタr2の値をI/Oプロセッサのローカルメモリ内のアドレスmem1に格納する命令である。

【００９０】以上の四つの命令は、読出先アドレス、書込先アドレスが共にI/Oプロセッサのローカルメモリであり、命令0-0がWait_Until_Next命令であることにより、I/Oプロセッサのローカルメモリを読出先アドレス－書込先アドレスとした命令が一つのスレッドに納められたことがわかる。図１５は、Wait_Until_Next命令の命令フォーマットの一例を示す。図１５において、本命令は図９に示したレジスタ指定型の算術演算命令の命令フォーマットを有する。本フォーマットの11bitから13bitを『010』に指定することにより、Wait_Until_Next命令のオペレーションをI/Oプロセッサは実行する。

【００９１】以上のように構成された第２実施形態のI/Oプロセッサによって、どのようにスレッドが割り当てられるかを図１７を参照しながら説明する。タスク(1)、タスク(2)、タスク(3)、タスク(4),タスク(5)には四サイクルのスレッドが割り当てられる。しかしタスク(0)には、wait_until_next命令が存在しているので、タスク(0)に割り当てられるスレッドは、wait_until_next命令が配置されている位置までとなり、タスク(0)に割り当てられるスレッドは２サイクルとなる。

【００９２】タスク(0)～タスク(5)が一巡した後に形成される次のフレームにおいて、タスク(1)は、他のタスクと同様４サイクルのスレッドが割り当てられる。以上のように本実施形態によれば、同一メモリをアクセス先としたメモリアクセス命令を一つのスレッドにおいて集中して実行することができる。

（第３実施形態）第１実施形態においては、非同期イベントタスクを起動すべき事象が成立したか否かに係らず、６本の非同期イベントタスクに公平にスレッドを割り当てていたので、起動する必要がない非同期イベントタスクと、その必要がある非同期イベントタスクとが実質同時間であるという悪平等が横行していた。第３実施形態は、非同期イベントタスクを起動すべき事象が成立しているか否かのチェックを非同期イベントタスクに行わせ、起動すべき事象が未成立のタスクについては、サイクルの割当数を削減するように制御を行う。

【００９３】第３実施形態におけるI/Oプロセッサの内部構成を図１８に示す。図１８においてI/Oプロセッサが命令メモリ１００、命令読出回路１０を備えている点は第１実施形態と変わりはない。第３実施形態において新規なのは、命令解読制御部１１及び演算実行部１４がそれぞれ、命令解読制御部８１及び演算実行部８４に置き換えられている点である。また７本の状態監視レジスタが追加され、タスク管理部１５の内部にタスク管理レジスタ１３が備えられた点である。

【００９４】七本の状態監視レジスタCR1～CR7は、それぞれが5ビット長のレジスタであり、AVデコーダ内部においてI/Oプロセッサの周辺に位置する構成要素、即ち、ビデオ出力部１０８、バッファメモリコントローラ１１０、RAMコントローラ１１１、FIFOコントローラ１１２等の構成要素と接続され、I/Oプロセッサの周辺部における５個の事象の成否が各ビットに反映されている（尚、ビデオ出力部１０８、バッファメモリコントローラ１１０、RAMコントローラ１１１、FIFOコントローラ１１２との接続は複雑であるため、その図示は省略している。）。

【００９５】その一例として状態監視レジスタCR1、状態監視レジスタCR2、状態監視レジスタCR3について図２３を参照しながら説明する。図２３は、状態監視レジスタCR1、状態監視レジスタCR2、状態監視レジスタCR3のビット構成を示した図である。状態監視レジスタCR1の第０ビットは、通常は『0』に設定され、バッファメモリバス１２１においてバッファメモリ１０２へのデータ転送が所定回数が行われたときのみ『1』に設定される。

【００９６】状態監視レジスタCR1の第１ビットは、通常は『0』に設定され、バッファメモリバス１２１に出力されたMPEGストリームのスタートコードが検出されたときのみ『1』に設定される。状態監視レジスタCR1の第２ビットは、通常は『0』に設定され、バッファメモリバス１２１に出力されたMPEGストリームに１バイト以上の有効データの存在が確認されたときのみ『1』に設定される。

【００９７】状態監視レジスタCR1の第３ビットは、通常は『0』に設定され、バッファメモリ１０２に、バッファに有効ビットが3Byte以上存在するときのみ『1』に設定される。また本ビットは、パージングタスクと、オーディオストリーム転送制御タスク、動画ストリーム転送制御タスクとの間の通信に用いられる。状態監視レジスタCR2の第０ビットは、通常は『0』に設定され、バッファメモリ１０２上でSDRAM３００への転送をサポートする旨の要求が発せられば『1』に設定される。また本ビットは、パージングタスクと、オーディオストリーム転送制御タスク、動画ストリーム転送制御タスクとの間の通信に用いられる。

【００９８】状態監視レジスタCR2の第１ビットは、通常は『0』に設定され、DMA転送が完了したときのみ『1』に設定される。状態監視レジスタCR2の第２ビットは、通常は『0』に設定され、ビットストリームFIFO１０３に対しての転送要求がある場合のみ『1』に設定される。状態監視レジスタCR2の第３ビットは、通常は『0』に設定され、FIFOコントローラ１１２内部の制御レジスタが更新されれば、『1』に設定される。状態監視レジスタCR3の第０ビットは、通常は『0』に設定さ

れ、バッファメモリ１０２がDMA_READY状態となれば『1』に設定される。

【００９９】状態監視レジスタCR3の第１ビットは、通常は『0』に設定され、SDRAM３００へのDMA転送が完了すれば『1』に設定される。状態監視レジスタCR3の第２ビットは、通常は『0』に設定され、外部に接続されたディスプレイ装置における水平帰線信号の立ち下がりにて『1』に設定される。

【００１００】状態監視レジスタCR3の第３ビットは、通常は『0』に設定され、外部に接続されたディスプレイ装置におけるビデオ信号が水平ブランキング期間になると『1』に設定される。状態監視レジスタCR3の第４ビットは、通常は『0』に設定され、ビデオアウトタスクに対する処理要求が発生した場合のみ『1』に設定される。

【００１０１】以上のように３本の状態監視レジスタは、AVデコーダ内の各バッファやコントローラの制御状態を表している。他の状態監視レジスタも同様であり、７本の状態監視レジスタによりI/Oプロセッサは、各非同期イベントタスクは、自身の起動要因となる各種事象が成立しているか否かを監視することができる。演算実行部８４は、演算実行部１４と同様の機能を有した構成である。演算実行部１４と比較して第１に新規なのは命令解読制御部８１からCR1～CR7といった各状態監視レジスタに付された番号と即値とを受け取り、指定された状態監視レジスタの保持値と受け取った即値との比較のための減算を行う点である。この減算結果に応じてゼロフラグ、キャリーフラグをオン／オフして状態監視レジスタの保持値と即値との一致、不一致、大小を命令解読制御部８１に知らせる。

【００１０２】上記の通知において命令解読制御部８１は現在実行中の演算を無効化する旨の信号を出力し得るが、これを受け取ると演算実行部８４は汎用レジスタへの値の格納を中断する。命令解読制御部８１は、命令解読制御部１１と同様の機能を有した構成であるが、命令解読制御部１１と異なるのは命令解読制御部８１は事象待ちループを形成し得る命令を命令メモリ１００から読み出して、これを解読した際にCR1～CR7といった各状態監視レジスタに付された番号と即値とを演算実行部８４に出力する。尚、事象待ちループを構成する命令とは、以下の｛例１｝の旨の内容である。状態監視レジスタの指定と即値とを演算実行部８４に出力した後、演算実行部８４がゼロフラグ、キャリーフラグにより保持値と即値との一致、不一致、大小を命令解読制御部８１に知らせると、命令解読制御部８１はこれを参照し、もし保持値と即値との不一致が通知されれば、タスク管理レジスタ１３に事象不成立を示す信号を出力し、この次の命令の実行結果を無効化する旨の信号を演算実行部８４に出力して読出先アドレスの値を先に進めないよう、命令読出回路１０に通知する。もし保持値と即値との不一致が通知されれば、タスク管理レジスタ１３に事象成立を示す信号を出力する。

｛例１｝

命令

事象iの成立の真偽を判定し、事象iの成立が偽ならば読出先アドレスを先には進めない。

【００１０３】また本命令のニーモニックを｛例２｝に示す。

｛例２｝

cmp_and_wait cr_statei,即値

｛例２｝において第１オペランドにcr_statei(i=0,1,2,3,4･･･7)が記述されているのは、I/Oプロセッサにおける７本の状態監視レジスタのうち何れかを指示できることを意味する。

【００１０４】この第１オペランドにおける状態監視レジスタの指定と、第２オペランドにおける即値指定とを組み合わせて、35もの事象のうち、任意のもの成否を確認することができる。上記命令を以降の説明ではCmp_And_Wait命令と呼ぶ。また上述した事象待ちループは、このCmp_And_Wait命令の第１オペランド、第２オペランドにおいて指定された事象の成否が何度も否と判定されることにより発生することはいうまでもない。

【００１０５】第３実施形態における変更が加えられた命令読出回路１０の構成を図１９に示す。図１９において命令読出回路１０がIF1+1保持部２０、increment回路２１、IF2保持部２２、DECPC保持部２３、タスク別PC格納部２４、セレクタ２５、セレクタ２６を備えている点は第１実施形態と差違はない。命令読出回路１０において新規なのは、DECPC保持部２３の出力段に接続されたフリップフロップ ２７である。

【００１０６】フリップフロップ ２７は、Cmp_And_Wait命令が指定した事象が不成立である場合に備え、命令解読制御部８１が解読中の命令の読出先アドレスをセレクタ２５に出力できるようDECPC保持部２３の値を保持している。もしプログラムカウント値を戻す旨の指示が命令解読制御部１１より与えられれば、フリップフロップ ２７はその保持値を⑤の信号線を通じてセレクタ２５に出力し、セレクタ２５は、このフリップフロップ ２７の保持値をタスク別PC格納部２４に格納するよう、タスク別PC格納部２４の入力元を⑤の信号線に切り換える。第３実施形態における命令読出回路の出力論理表を図２０に示す。

【００１０７】図２６は、Cmp_and_Wait命令の解読時のフレームにおけるI/Oプロセッサの構成要素についてのタイミングチャートであり、本タイミングチャートにおけるフリップフロップ２７の役割について説明する。図２６においてフリップフロップ ２７が保持値としてCmp_and_Wait命令のアドレスに相当する命令0-0のアドレスを参照符号a7に示すように出力すると、命令0-0のアドレスはIPC0としてタスク別PC格納部２４に格納される。こ

れによりタスク別PC格納部２４には、IPCOとしてCmp_and_Wait命令のアドレスが格納されることになる。

【０１０８】本実施形態におけるタスク管理部１５の内部構成を図２１に示す。図２１に示すタスク管理レジスタ１３は、事象待ちループ（以降wait状態と呼ぶ）に陥ったタスクを個別に管理するためのビットが割り当られたレジスタである。タスク管理レジスタ１３は命令解読制御部８１から事象不成立を示す信号が通知されると、タスクID保持部７２が保持しているタスク識別子taskidを参照し、そのタスク識別子taskidに対応するビットを『1』に切り換える。逆に事象成立を示す信号が通知されると、タスクID保持部７２が保持しているタスク識別子taskidを参照し、そのタスク識別子taskidに対応するビットを『0』に切り換える。

【０１０９】図２２は、タスク管理レジスタ１３におけるwait状態タスクの管理用のビット構成を示す。本図において第０ビットが『1』であれば、タスク(0)がwait状態であることを示し、第０ビットが『0』であればタスク(0)がそうでないことを示す。また本図において第１ビットが『1』であれば、タスク(1)がwait状態であることを示し、第１ビットが『0』であればタスク(1)がそうでないことを示す。タスク管理レジスタ１３の第２ビットが『1』であれば、タスク(2)がwait状態であることを示し、第２ビットが『0』であればタスク(2)がそうでないことを示す。タスク管理レジスタ１３による事象待ちループの管理により、複数のタスクにおいて同時多発した事象待ちループを個別に管理することができる。

【０１１０】図２６のタイミングチャートにおいて、命令0-0がCmp_and_Wait命令であると解読され、事象が不成立であると判定されたものとする。この場合、命令解読制御部８１はタスク(0)が事象成立待ち状態になった旨を参照符号a8に示すように通知する。そうすると、タスク管理レジスタ１３はタスク(0)のタスク識別子に対応するビットを参照符号a9に示すように『1』に設定する。

【０１１１】続いて『事象成立待ち時』、『事象成立時』、『事象成立時以降』においてタスク管理レジスタ１３に割り当てられた各ビットがどのように設定されるかについてタイミングチャートを参照しながら説明する。図２７に事象成立待ち時、図２８に事象成立時、図２９に事象成立時以降におけるタイミングチャートを示す。

【０１１２】タスク(0)におけるCmp_and_Wait命令が成立を待っていた事象が図２８において成立したとする。この場合、命令解読制御部８１は事象が成立した旨を参照符号d1に示すように通知する。そうすると、タスク管理レジスタ１３はタスク(0)のタスク識別子に対応するビットを参照符号d2に示すように『0』に設定する。図２１を参照して、第３実施形態におけるスレッドマネージャー６１の内部構成について説明する。スレッドマネージャー６１においてフリップフロップ ５１、カウンタ５２、比較器５４を備えている点は第１実施形態と差違は無い。図２１において新規なのはIDコンバータ５３及びセレクタ５５である。

【０１１３】IDコンバータ５３は、タスクID保持部７２に次にスレッドを割り当てるべきタスクの識別子が出力されると、タスク管理レジスタ１３においてwait状態に設定されているタスクと次にスレッドを割り当てるべきタスク識別子taskidとが一致しているかを判定し、もし一致していればセレクタ５５にHigh値を出力し、不一致ならばセレクタ５５にLow値を出力する。

【０１１４】セレクタ５５は、IDコンバータ５３が出力した信号のHigh値、Low値に応じて、『2』或は『4』の数値を択一式に比較器５４に出力する。図２６において次にスレッドを割り当てるべきタスクがタスク(0)なら、参照符号a10においてセレクタ５５にLow値が出力されるので参照符号a12に示すように、セレクタ５５は『4』を比較器５４に出力する。しかし図２６の参照符号a9においてタスク(0)についてのビットが『1』に設定されて、図２７の参照符号g1に示すようにセレクタ５５にHigh値が出力されると、セレクタ５５は参照符号g2に示すように『2』を比較器５４に出力する。

【０１１５】図２８の参照符号d2においてタスク(0)についてのビットが『0』に設定されて、図２９において参照符号g5に示すようにセレクタ５５にHigh値が出力されると、セレクタ５５は参照符号g6に示すように『4』を比較器５４に出力する。　セレクタ５５が『2』又は『4』の数値を択一的に比較器５４に出力することにより、２サイクルのスレッド、４サイクルのスレッドの何れか一方が各タスクに割り当てられる。セレクタ５５が『2』を出力した際には、その第２番目の命令を無効化するので、その結果、１サイクルのスレッドが割り当てられることになる。

【０１１６】タイミングチャートの時間軸に沿って、図２６から図２９までのタイミングチャートに示されている命令読出回路１０の構成要素及びタスク管理部１５の構成要素の動作について説明する。初期状態においてタスク管理レジスタ１３のビットはオールゼロであるものとする。そうして図２６においてタスク(0)から命令の読み出しが開始されると、タスク(0)についてのビットはタスク管理レジスタ１３において『0』に設定されているので、参照符号a12に示すようにセレクタ５５はカウンタの上限値として『4』を出力する。カウンタの上限値が『4』に設定されたので、タスク(0)を四サイクル実行しようとするが、命令0-0がCmp_and_Wait命令であるので、参照符号a7に示すようにタスク別PC格納部２４に命令0-0の読出先アドレスを書き込む。それと共に、参照符号a8,a9に示すようにCmp_and_Wait命令の実行時においてタスク管理レジスタ１３の第０ビットを『1』に設定する。この設定により、タスク(0)は事象成

立待ちとなる。このようにビットを設定した後、タスク(1)、タスク(2)、タスク(3)、タスク(4)、タスク(5)の実行が済んで次のフレームに移行したものとする。

【０１１７】事象成立待ち状態においてタスク管理レジスタ１３のビットはタスク(0)のみが『1』であり、その他のビットはゼロであるものとする。そうして図２７においてタスク(0)から命令の読み出しが開始されると、タスク(0)についてのビットはタスク管理レジスタ１３において『1』に設定されているので、参照符号g2に示すようにセレクタ５５はカウンタの上限値として『２』を出力する。カウンタの上限値が『２』に設定されたので、タスク(0)を２サイクル実行しようとするが、命令0-0がCmp_and_Wait命令であるので、タスク別PC格納部２４に命令0-0の読出先アドレスを書き込む。このような書き込み後、タスク(1)、タスク(2)、タスク(3)、タスク(4)、タスク(5)の実行が済んで次のフレームに移行したものとする。

【０１１８】次のフレームにおいて事象が成立したものとする。図２８においてタスク(0)から命令の読み出しが開始されると、タスク(0)についてのビットはタスク管理レジスタ１３において『1』に設定されているので、参照符号g4に示すようにセレクタ５５はカウンタの上限値として『２』を出力する。カウンタの上限値が『２』に設定されたので、タスク(0)を２サイクル実行しようとする。

【０１１９】命令0-0であるCmp_and_Wait命令の実行時において、事象成立が確認されるので、参照符号d1,d2に示すようにCmp_and_Wait命令の実行時においてタスク管理レジスタ１３の第０ビットを『0』に設定する。この設定により、タスク(0)は事象成立待ちから解放される。このようにビットを設定した後、タスク(1)、タスク(2)、タスク(3)、タスク(4)、タスク(5)の実行が済んで次のフレームに移行したものとする。

【０１２０】図２９に示す事象成立以降のフレームにおいては、タスク(0)についてのビットがタスク管理レジスタ１３において『0』に設定されているので、参照符号g6に示すようにセレクタ５５はカウンタの上限値として『４』を出力する。カウンタの上限値が『４』に設定されたので、タスク(0)を４サイクル実行する。以上のように構成された第３実施形態のI/Oプロセッサによって、どのようにスレッドが割り当てられるかを図２４（ａ）～図２４（ｄ）を参照しながら説明する。

【０１２１】図２４（ａ）においてタスク(0)、タスク(3)、タスク(4)、タスク(5)は何れもメモリ転送制御タスクであり、その第１番目の命令がCmp_and_Wait命令であるものとする。ここで図２４（ｂ）の状態では、これらのCmp_and_Wait命令が待機している事象が未成立であるものとすると、タスク(0)、タスク(3)～タスク(5)に２サイクルのスレッドが割り当てており、第２命令は無効化されているが、タスク(1)、タスク(2)には４サイクルのスレッドが割り当てられる。

【０１２２】図２４（ｃ）においてタスク(0)～タスク(5)が何回か巡回し、タスク(0)のCmp_and_Wait命令が待っていた事象のみが成立したものとする。そうすると当該事象が成立したフレームにおいては、タスク(3)～タスク(5)には依然として2サイクルのスレッドが割り当てられており、第２命令は無効化されているが、タスク(0)は事象が成立したので第２命令は実行される。

【０１２３】図２４（ｄ）において事象j0の成立以降のフレームにおいては、タスク(0)には、タスク(1)、タスク(2)と同一の４サイクルのスレッドが割り当てられる。図２５は、Cmp_and_Wait命令の命令フォーマットの一例を示す。図２５において、5bitから7bitまでの3bitを用いて7本の状態監視レジスタのうち何れかを指定でき、11bitから15bitまでを用いて 5bit長の即値を格納できる。本フォーマットの9bitから10bitを用いてオペレーション内容の指定が可能である。この2bitを『10』に指定することにより、Cmp_and_Wait命令のオペレーションをI/Oプロセッサは実行する。また、この2bitを『00』に指定することにより、七本の状態監視レジスタのうち、一本のものの値と5bit即値との比較命令のオペレーションをI/Oプロセッサは実行する。

【０１２４】以上のように本実施形態によれば、メモリアクセスが可能な状態を待っているメモリ転送制御タスクにより短い時間長のスレッドを割り当てることにより、他のタスクの進捗を向上させることができる。

（第４実施形態）第４実施形態は、ディスプレィの水平ブランキング期間、垂直ブランキング期間において、ビデオアウトタスクにより多くのスレッドを割り当てる技術に関する。

【０１２５】スレッドの増加を行うI/Oプロセッサは、第１実施形態におけるタスク管理部１５の構成に図３０～図３１に示す変更を加えて実現される。第４実施形態におけるI/Oプロセッサは、命令メモリ１００、命令読出回路１０、命令解読制御部８１、演算実行部８４、タスク管理部１５を備えている点は第３実施形態と変わりはない。第４実施形態において新規なのは、タスク管理レジスタ１３が緊急状態遷移許可信号iexecmodeを監視している点である。

【０１２６】第４実施形態におけるタスク管理部１５の内部構成を図３０に示す。図３０におけるタスク管理レジスタ１３における3ビットは、タスク管理レジスタ１３内の3ビットを用いて緊急扱いを要するタスクの識別子が指定される。当該3ビットにおいて六個のタスクのうちどのタスクを緊急扱いするかは、上記緊急状態遷移許可信号iexecmodeがどのような内容の信号であるかに応じて適宜変更する。本実施形態では、緊急状態遷移許可信号iexecmodeは水平ブランキング期間、垂直ブランキング期間等に相当し、ビデオアウトタスクの識別子をタスク管理レジスタ１３における3ビットに指定してい

る。

【０１２７】何れかのタスクを緊急扱いを要するタスクとしてタスク管理レジスタ１３に登録するかは、何れのタスク内にemergecy命令が含まれているかをタスク管理レジスタ１３が参照することにより、決定されるものとする。命令解読制御部８１が次に実行すべき命令を解読した際、その解読結果がemergecy命令であれば、タスク管理レジスタ１３は当該命令を含むタスクを緊急扱いを要するタスクとしてその識別子を保持する。

【０１２８】第４実施形態におけるスケジューラ６２の内部構成を図３１に示す。図３１においてスケジューラ６２がタスクID保持部７２を備えている点は第１実施形態と差違は無い。図３１において新規なのは、タスクラウンド管理部７３及びプライオリティエンコーダ７４がそれぞれ、タスクラウンド管理部７５及びプライオリティエンコーダ８０に置き換えられている点である。またコンバータ７６、緊急タスクマスク部７７及びタスクID保持部８３が追加されている点である。

【０１２９】タスクラウンド管理部７５がタスクラウンド管理部７３と異なるのは、タスクラウンド管理部７３がタスクID保持部７１においてタスク識別子がどのように更新されてきたかを監視し、6ビット長のレジスタを用いて六個のタスクのうちスレッドの割り当てが済んだものを管理するのに対して、タスクラウンド管理部７５は、スレッドの割り当てが済んだタスクの管理を、通常期間におけるタスク管理と、緊急期間における通常タスクの管理とに区別して行う点である。例えばタスク管理レジスタ１３においてタスク(1)をインターリーブ式に実行する場合、６個のタスクのうち５個のタスクがタスク(0)、タスク(2)、タスク(3)、タスク(4)、タスク(5)というように巡回することになる。

【０１３０】５個のタスクを上記のように巡回する場合、実行の済んだタスクを示すラウンド値はタスク管理レジスタ１３において管理されている緊急タスク抜きの数値で表現する必要がある。このようにタスクラウンド管理部７５は緊急タスク抜きのラウンド値を管理するため、ラウンド値taskn とは別に緊急タスク抜きのラウンド値taskneを緊急タスクマスク部７７に出力する。ちなみにtaskneの『ne』とは、『Not』及び『Emergency』の頭文字に由来している。

【０１３１】コンバータ７６は、タスク管理レジスタ１３に管理されている緊急タスクの識別子を6ビットのラウンド値emgcytaskに変換する。緊急タスクマスク部７７は、コンバータ７６により変換されたラウンド値emgcytaskと、タスクラウンド管理部７５が出力したラウンド値taskneとの6ビット長の論理和をとるOR回路７８を備える。

【０１３２】ここでOR回路７８において論理和をとるのは、ラウンド値taskne及び緊急タスクを示すラウンド値の両者を含んだ6ビット長のラウンド値を算出すること

を意味する。例えばラウンド値taskneがタスク(0)、タスク(2)、タスク(3)にスレッドが割り当てられたことを示す『001101』であり、ラウンド値emgcytaskが緊急タスクがタスク(1)であることを示す『000010』である場合、OR回路７８の出力は、タスク(0)、タスク(1)、タスク(2)、タスク(3)にスレッドが割り当てられたことを示す『001111』となる。

【０１３３】プライオリティエンコーダ８０は、プライオリティエンコーダ７４の機能に加えて、タスクラウンド管理部７５が出力したラウンド値taskne（ラウンド値maskne）を受け取り、受け取ったタスクラウンド管理部７３のラウンド値taskne（ラウンド値maskne）において何ビット目に『1』から『0』への反転が生じているかを検出する。このように『0』への反転が生じたビットを検出すると、この反転が生じたビットに割り当られているタスクのタスク識別子taskidをそれぞれ個別にタスクID保持部８３に格納させる。

【０１３４】例えば緊急タスクのラウンド値とラウンド値taskneとの論理和であるラウンド値maskne『000111』が緊急タスクマスク部７７から出力されると、プライオリティエンコーダ８０はこのラウンド値maskneにおいて下位から第３ビット目に『0』への反転が生じていることを検出する。この第３ビット目は図７（ｂ）においてタスク(3)に割り当られているビットであるから、プライオリティエンコーダ８０はタスク(3)のタスク識別子taskidをセレクタ８２に出力する。

【０１３５】cstate記憶部８５は、I/Oプロセッサが通常状態であるか緊急状態であるかを示すフラグcstateを記憶する。図３３は、cstate記憶部８５が記憶するフラグの状態遷移図である。本図において、通常状態から緊急状態への遷移は、緊急状態遷移許可信号iexecmodeがHighであり、尚且つタスク切替信号chg_task_exがHigh値になった場合に行われる。逆に緊急状態から通常状態への遷移は、タスク切替信号chg_task_exがHigh値になった場合に行われる。図３４は緊急状態遷移許可信号iexecmode、タスク切替信号chg_task_ex、cstate間のタイミングを示すタイミングチャートである。

【０１３６】iexecmode信号がHighに立ち上がったタイミングa30において参照符号a31に示すようにタスク切替信号chg_task_exがHigh値になったとする。そうするとcstateは参照符号a32に示すようにHighとなって緊急状態に遷移する。タスク切替信号chg_task_exが参照符号a33に示すように再度Highになるとcstateは参照符号a34に示すようにLow値となって通常状態に戻る。このような緊急状態から通常状態への遷移、通常状態から緊急状態への遷移を繰り返すことになる。

【０１３７】セレクタ８２は、入力されてきた緊急状態遷移許可信号iexecmodeがHighであり、尚且つcstateが参照符号a32に示すようにHighである場合、（cstateの逆の真理値cstate!が参照符号a35に示すようにLowであ

る場合）に、タスク管理レジスタ１３に記憶されているタスク識別子taskideを選択してタスクID保持部７２に出力する。その他の場合、プライオリティエンコーダ８０の出力であるラウンド値maskneから変換されたタスク識別子taskidneを選択してタスクID保持部７２に出力する。図３２は、セレクタ８２におけるtaskideの真理値表を示す図である。

【０１３８】このようにラウンド値maskneから変換されたタスク識別子taskidneと、緊急タスクの識別子とをセレクタ８２が緊急状態遷移許可信号iexecmodeのHigh期間／Low期間に応じて切り換えて出力することは、緊急状態遷移許可信号iexecmodeのHigh期間／Low期間に応じて、緊急タスク抜きの何れかのタスクと、緊急タスクとを切り換えてタスクID保持部７２に格納することを意味する。

【０１３９】例えばタスク(1)が緊急タスクであるものとすると、緊急タスク抜きのタスク識別子として、タスク(0)、タスク(2)、タスク(3)、タスク(4)、タスク(5)というようにセレクタ８２はタスク識別子を出力する。通常状態では、図３５（ａ）に示すように、セレクタ８２により緊急タスク抜きの何れかのタスクが出力されるため、タスク(0)、タスク(2)、タスク(3)、タスク(4)、タスク(5)というように、タスク(1)に割り当てられるスレッドは０サイクルとなる。

【０１４０】緊急状態では、図３５（ｂ）に示すように、セレクタ８２により緊急タスク抜きの何れかのタスクと、緊急タスクとが選択的に出力されるため、タスク(0)、タスク(1)、タスク(2)、タスク(1)、タスク(3)、タスク(1)、タスク(4)、タスク(1)、タスク(5)、タスク(1)というように、タスク(1)は二回に一回の割合でスレッドが割り当てられる。セレクタ８２による緊急状態におけるタスク識別子の出力では、タスク(1)にタスク実行が巡ってくるのは二回につき一回となり確率に換算すると１／２となる。

【０１４１】このような１／２の確率は、通常の実行確率の３倍という極めて高確率であり、緊急タスクは非常に高速に実行されることになる。以上のように本実施形態によれば、水平ブランキング期間、垂直ブランキング期間においてビデオアウトタスクのみを優先して実行することができる。この期間において動画データを映像信号に変換すれば、映像信号をディスプレィの表示期間までに好適に処理することができる。

【０１４２】尚第４実施形態の構成では、緊急タスクを優先して実行するため、緊急タスク以外のタスクの実行頻度が落ちてしまうという弊害が発生するが、これを防止する防止機構を設けても良い。その防止機構とは、『最低限これだけの実行速度を維持しなければならない』という制約付きタスク（例外タスクという）の識別子をタスク管理レジスタに記憶させ、この例外タスクについては、四命令実行というタスク切り換えの条件を除外する旨の信号をスレッドマネージャー６１宛に出力させる。

【０１４３】スレッドマネージャー６１内の比較器５４は、この旨の信号を受け付けると、タスク切替信号の出力を所定時間だけ猶予する。ここでの所定時間の時間長とは緊急タスク優先による各タスクの遅れを取り戻せるだけの時間長である。このように所定時間だけタスク切替信号の出力を猶予することにより、例外タスクの処理を４命令以上行わせ、緊急タスク挿入に伴う処理の遅延を取り戻させることができる。

【０１４４】また本実施形態は、外部から入力されてくる特定信号に同期して、特定タスクを高速化に実行することについて述べているが、特定命令によって高速実行するタスクを定めても良い。状態監視レジスタCR2の第３ビットは、通常は『0』に設定され、外部に接続されたディスプレィ装置におけるビデオ信号が帰線期間になると『1』に設定されるので、状態監視レジスタCR2と、即値とをオペランドに指定し、これらの一致、不一致を判定させる旨のCMP命令をビデオアウトタスクに配置しておき、この命令についての演算の実行結果が真となる場合のみ、ビデオアウトタスクを高速に実行してもよい。命令のオペランドにて緊急扱いを要するタスクを指定できるようにしてもよい。

【０１４５】更に通常状態にて実行すべきタスクを緊急タスク抜きのラウンド値taskneにて指定したが、通常状態にて緊急タスクを実行してもよい。

（第５実施形態）第５実施形態は、タスク同士の連携により、タスクの巡回効率を向上させることを意図した実施形態である。

【０１４６】ここでタスク同士の連携とは、特定時期が到来しないとスレッドの割り当てが無駄に終わるタスクは自ら進んで休眠状態に入り、休眠状態の解除を他のタスクに委ねる。そして解除を委ねられたタスクは、その休眠状態を解除すべき時期が到来するのを待ち、もし到来すれば特定タスクの休眠状態を解除することをいう。

【０１４７】タスクの休眠は、複数のタスクにおいて同時多発する場合もあるし、全く発生しない場合もある。休眠状態に陥ったタスクを個別に管理できるよう、第５実施形態ではタスク管理レジスタ１３に休眠（sleep状態）に入ったタスクを管理するためのビットを割り当てている。図３６は第５実施形態におけるタスク管理部１５の構成であり、図３７はタスク管理レジスタ１３のビット構成及び、これらの各ビットがどのようなフラグとして機能するかを示す説明図である。本図においてタスク管理レジスタ１３の第０ビットが『0』であれば、タスク(0)をsleep状態として処理することを示し、第０ビットが『1』であればタスク(0)がそうでないことを示す。

【０１４８】タスク管理レジスタ１３の第１ビットが『0』であれば、タスク(1)をsleep状態として処理する

ことを示し、第１ビットが『1』であればタスク(1)がそうでないことを示す。タスク管理レジスタ１３の第２ビットが『0』であれば、タスク(2)をsleep状態として処理することを示し、第２ビットが『1』であればタスク(2)がそうでないことを示す。これらのビットをタスクが切り換えることにより、自身を休眠状態にしたり、他のタスクの休眠状態を解除したりする。

【０１４９】タスク管理レジスタ１３のビット操作は、以下の〔例３〕〔例４〕にニーモニックを示す命令によりなされる。

〔例３〕

sleep_task

〔例４〕

wake_task task(k)

〔例３〕において『sleep_task命令』は、自身をsleep状態にする命令である。〔例４〕において『wake_task命令』は、第１オペランドにより６つのタスクのうち何れか一つの識別子を指定できる。このような二種の命令を各タスク内に配置することにより、各タスクは休眠状態の移行と、これの解除とを交互に行う。尚、sleep_task命令においては第１オペランドにより他のタスクを指定できるようにしてもよい。

【０１５０】図３８は、第４実施形態におけるスケジューラ６２の内部構成を示す図である。図３８においてスケジューラ６２がタスクID保持部７１、タスクID保持部７２、タスクラウンド管理部７３、プライオリティエンコーダ７４を備えている点は第１実施形態と差違は無い。本図において新規なのはインバータ９４及びOR回路９５からなるタスクスキップ管理回路９３である。

【０１５１】タスクスキップ管理回路９３は、タスク管理レジスタ１３においてsleep状態に割り当られている6ビットを反転するインバータ９４と、反転された6ビットと、タスクラウンド管理部７３から出力されたラウンド値taskn との論理和をとるOR回路９５からなる。ここでタスク(0)、タスク(1)、タスク(2)の実行が済んで『000111』というラウンド値taskn がタスクラウンド管理部７３から出力され、またタスク管理レジスタ１３においてタスク(3)がsleep状態に設定されているものとする。この場合タスク管理レジスタ１３のsleepビットは『110111』となり、インバータ９４によりこれの反転値『001000』が出力される。インバータ９４による反転値『001000』とラウンド値taskn『000111』との論理和は、『001111』となる。この論理和がプライオリティエンコーダ７４に出力されると、第４ビットにおいて『1』から『0』への反転が行われていることが検出され、タスク(4)の識別子がタスクID保持部７１及びタスクID保持部７２に出力される。タスク(4)の識別子が出力されると、タスク(0)、タスク(1)、タスク(2)、タスク(4)の順に実行が行われることになる。

【０１５２】図３９は、sleep命令を命令1-1,2-1,3-1としてその内部に有した非同期イベントタスクの一例であり、図４１は、図３９の非同期イベントタスクがどのように実行されるかを示すタイミングチャートである。sleep命令が図４１に示すように命令1-1として非同期イベントタスクの第１番目の命令として格納されている場合、この命令1-1がDECPC保持部２３に格納され、命令解読制御部１１により解読されると、セレクタ２５は、①の信号線を介して命令1-2の読出先アドレスをタスク別PC格納部２４に格納する。その後命令解読制御部１１は、命令1-2、命令1-3を無効化する（図１６における『NOP』）。

【０１５３】図４０はsleep命令及びwake命令の命令フォーマットの一例を示す。図４０において、本命令は図９に示したレジスタ指定型の算術演算命令の命令フォーマットを有する。本フォーマットの11bitから13bitを『010』に指定することにより、sleep命令のオペレーションをI/Oプロセッサは実行する。また本フォーマットの11bitから13bitを『011』に指定することにより、wake命令のオペレーションをI/Oプロセッサは実行する。

【０１５４】以上のように構成された第５実施形態のI/Oプロセッサによって、どのようにスレッドが割り当てられるかを図４２（ａ）～図４２（ｄ）を参照しながら説明する。タスク(1)はパージングタスク、タスク(3)はホストI/Oタスクであり、タスク(1)、タスク(2)、タスク(3)の第１番目の命令がsleep命令であるものとする。タスク(0)、タスク(4)、タスク(5)に４サイクルのスレッドが割り当てられるが、図４２（ａ）においてタスク(1)、タスク(2)、タスク(3)には、第１番目の命令としてsleep命令が格納されているので2サイクルのスレッドが割り当てられる。

【０１５５】sleep命令の実行後、次のフレームにおいては図４２（ｂ）に示すように、タスク(1)、タスク(2)、タスク(3)に割り当てられるスレッドは０サイクルである。タスク(0)は起動期間がより長いビットストリーム転送制御タスクであり、タスク(0)には第１番目にタスク(3)に対してのwake命令が格納されているものとする。そうすると、図４２（ｃ）に示すように次のフレームにおいてこのwake命令が解読されることによりタスク(3)に４サイクルのスレッドが割り当てられる。

【０１５６】タスク(0)には第１４番目にタスク(1)に対してのwake命令が格納されているものとする。そうすると、図４２（ｄ）に示すように次のフレームにおいてこのwake命令が解読されることによりタスク(1)に４サイクルのスレッドが割り当てられる。以上のように本実施形態によれば、パージングタスク、ホストI/Oタスク等、起動すべき期間が限定されているタスクは自ら進んで休眠状態に入り、起動期間がより長い転送制御タスクに休眠状態の解除を委ねることにより、タスク巡回のスループットを向上させることができる。

（第６実施形態）第６実施形態は、第３実施形態におけ

るwait状態制御と、第４実施形態における緊急タスク制御と、第５実施形態における休眠タスク制御とを共存させる技術に係る。

【０１５７】図４３は、タスク管理レジスタ１３の各ビットにwait状態、emgcy状態、sleep状態の管理を統括させたものである。スレッドマネージャー６１及びスケジューラ６２は、このようにタスク管理レジスタ１３に統括されたwait状態、emgcy状態、sleep状態を参照して次のタスクの決定やタスクの次スレッドに割り当る命令数の増減を行う。

【０１５８】図４４は、緊急タスク制御及びsleep状態タスク制御を共存させ得るよう構成されたスケジューラ６２の構成図である。図４４においてスケジューラ６２がタスクID保持部７２、タスクラウンド管理部７５、コンバータ７６、緊急タスクマスク部７７、プライオリティエンコーダ８０、セレクタ８２、タスクID保持部８３から構成されている点は第４、第５実施形態と差違は無い。本図において、タスクラウンド管理部７５の出力と、インバータ９４により反転されたラウンド値taskneのラウンド値との論理和をとるOR回路９６が新たに追加され、その結果をプライオリティエンコーダ８０に出力する。

【０１５９】プライオリティエンコーダ８０は、OR回路９６が出力したラウンド値をタスク識別子に変換して、タスクID保持部７２及びタスクID保持部８３に出力する。以上のように本実施形態によれば、第４実施形態における緊急タスク制御と、第５実施形態におけるsleep状態タスク制御とを共存させることにより、より柔軟性に富んだタスクスケジューリングを実現することができる。

（各実施形態における全体制御についての説明）以降、第１実施形態～第５実施形態に示したI/Oプロセッサの全体制御をフローチャートを参照しながら説明する。

【０１６０】図４５は、第１実施形態におけるI/Oプロセッサの全体制御を図示したフローチャートである。本図において、『変数k』とは命令メモリ１００に記憶されている６本の非同期イベントタスクを指示するための変数でありタスクID保持部７１及びタスクID保持部７２が保持するタスク識別子taskidに対応する。『変数i』とはそれぞれのタスクに含まれる命令の実行の際、その命令が当該スレッドにおいて何番目に実行されたかを指示するための変数であり、カウンタ５２によりカウントされるカウント値に対応する。『変数adr』とは、タスク別PC格納部２４に格納されている個々のタスクの読み出し先アドレスを示す変数である。

【０１６１】図５０は、図４５のフローチャートによるスレッド間の遷移を示す図である。図５０において、縦方向下向きが時間軸であり、４つの長方形を含む太枠で囲まれた四角形がスレッドである。図４５のフローチャートにおける変数kにより個別に指示される。太枠の中の一つの各四角形は一つの命令を実行するサイクルを表し、これらの命令は変数iにより指示される。ステップＳ１は、上記タスクID保持部７１がゼロのタスク識別子taskidを出力する。これにより変数kがゼロクリアされ、ステップＳ２に移行する。ステップＳ２は、カウンタ５２を初期化することにより変数iを１に初期化する。ステップＳ３においてタスクID保持部７２がタスク別PC格納部２４に読み出しアドレス選択信号nxttaskid(rd_adr)を出力することによりタスク(k)の読出先アドレスをタスク別PC格納部２４から読み出す。ステップＳ４においてタスク(k)の読出先アドレスを用いて、命令メモリ１００から命令を取り出す。ステップＳ５において命令解読制御部１１は、取り出された命令を解読して実行する。

【０１６２】ステップＳ６は、比較器５４は変数iと上限4との比較を行う。この場合はi=1であるのでステップＳ７に移行する。ステップＳ７は、カウンタ５２に変数iのインクリメントを行わせ、increment回路２１に読出先アドレスのインクリメントを行せるステップであるが、このインクリメントはタスクkにおけるある命令の実行から別の命令の実行への切り換えを意味する。この場合i=2であるから、スレッドにおける命令の実行から第２命令の実行への切り換えが行われることになる。

【０１６３】変数iは、ステップＳ７においてインクリメントされてステップＳ４に移行する。変数iのインクリメントよりi=2となり、読出先アドレスも一命令分進行したので、ステップＳ４において命令読出回路１０は命令メモリ１００内の読出先アドレス(adr+1)で指示される命令を読み出して、ステップＳ５において命令解読制御部１１はこれを解読し、実行する。

【０１６４】以上のステップＳ４～ステップＳ７の一連の手順を繰り返し行うことにより変数iは次々とインクリメントされ、命令メモリ１００におけるadr+0,adr+1,adr+2,adr+3の領域に記憶されている命令を順次読み出す。以上のステップＳ４、ステップＳ５の実行をステップＳ６においてYesとなるまで繰り返す。ステップＳ６がYesとなると、ステップＳ６からステップＳ８への移行が行われる。以上の手順において、変数iが1,2,3である内はステップＳ４、ステップＳ５、ステップＳ６、ステップＳ７の手順が繰り返されて、変数iが4になって初めてステップＳ６からステップＳ８への移行が行われたことは、タスク(0)に対する命令実行を４回繰り返すことを意味する（図５０において最上段に位置する４つの四角形の並びはタスク(0)の第１スレッドにおける４回の命令実行の図解となる。）。ちなみにステップＳ８～ステップＳ１０の手順はタスクからタスクへの切り換えを行うものであるから、ステップＳ６における『変数i＝４』の判定は、命令の実行の４回の繰り返しをタスク切り換えの条件に課していることを意味する。

【０１６５】ステップＳ６において変数iと上限値との

判定がなされると、ステップＳ８においてincrement回路２１により変数iがインクリメントされ、ステップＳ９においてタスク(k)の読出先アドレス(adr+i-1)を読出先アドレス(adr)としてタスク別PC格納部２４に格納した後にステップＳ１０に移行する。ステップＳ１０では、タスクラウンド管理部７３及びプライオリティエンコーダ７４に次に実行すべきタスクを決定させる。この際タスクラウンド管理部７３により『000000』のラウンド値taskが出力され、プライオリティエンコーダ７４により、『001』のタスク識別子taskidが出力される。『001』のタスク識別子taskid、即ちk=1となるので、タスク(0)からタスク(1)への切り換えが行われる。タスクの切り換え後、ステップＳ２に移行すると、変数iの初期化がなされ再度ステップＳ３～ステップＳ７の繰り返しがまた開始される。

【０１６６】前回同様ステップＳ４において命令メモリ１００における当該読出先アドレスからタスク(1)に含まれる一命令を取り出し、ステップＳ５において取り出された命令を解読して実行する。ステップＳ６は、変数iと上限4との比較を再度行う。この場合はi=1であるのでステップＳ７に移行する。変数iは、ステップＳ７においてインクリメントされてステップＳ４に移行する。変数iのインクリメントよりi=2となるので、ステップＳ４において読出先アドレス(adr+1)で指示される命令メモリ１００内の領域のアクセスを行い、命令メモリ１００において、読出先アドレス(adr+1)で指示される領域に記憶されている命令を読み出して、ステップＳ５においてこれを解読し、実行する。

【０１６７】以上のステップＳ４～ステップＳ７の一連の手順を繰り返し行うことにより変数iを次々とインクリメントして、命令メモリ１００におけるadr+0,adr+1,adr+2,adr+3の領域に記憶されている命令を順次読み出す。以上のステップＳ４、ステップＳ５の実行をステップＳ６においてYesとなるまで繰り返す。ステップＳ６がYesとなると、ステップＳ６からステップＳ８への移行が行われる。以上の手順において、変数iが1,2,3である内はステップＳ４、ステップＳ５、ステップＳ６、ステップＳ７の手順が繰り返されて、変数iが4になって初めてステップＳ６からステップＳ８への移行が行われる（以上までの動作により、図５０ではタスク(1)における四命令分（タスク(1)に対する４個の命令）進行したことになる。）。ステップＳ８において変数iをインクリメントして、ステップＳ９においてタスク(k)の読出先アドレス(adr+i-1)を読出先アドレス(adr)として格納した後にステップＳ１０に移行する。ステップＳ１０において再度タスクラウンド管理部７３及びプライオリティエンコーダ７４に変数kの決定行わせて、変数kをk=2にして、タスク(1)からタスク(2)への切り換えを行わせる。タスクの切り換え後、ステップＳ２に移行すると、変数iの初期化がなされ再度ステップＳ３～ステップＳ７の繰り返しがまた開始される。

【０１６８】以上の繰り返しにより、図５０において、タスク(1)、タスク(2)、タスク(3)、タスク(4)、タスク(5)の実行が四命令ずつ行われ、各タスクが順次消費されてゆく。

（第２実施形態における全体制御の説明）第２実施形態におけるI/Oプロセッサの全体制御は、図４６のフローチャートに示す手順にて行われる。図４６のフローチャートは、ステップＳ１～ステップＳ１０からなる図４５のフローチャートをベースにして作成されている。ここで図４６において新規なのはステップＳ１２がステップＳ４～ステップＳ５の間に挿入されている点である。

【０１６９】ステップＳ１２は、ステップＳ４により命令メモリ１００から読み出された命令がWait_Until_Next命令であるかを判定する。もしWait_Until_Next命令であれば、ステップＳ８へと移行して変数iをインクリメントし、ステップＳ９においてタスク(k)の読出先アドレス(adr+i-1)を読出先アドレス(adr)として格納する。

【０１７０】ここで留意すべきは、ステップＳ１２では変数iの値に拘らず、Wait_Until_Next命令が実行された時点にステップＳ８に移行して読出先アドレスを格納する点である。このようにステップＳ８が変数iの値に拘らずステップＳ１２の解読後に実行されるため、タスクからタスクへの切り換えは、四回の命令を待たずして行われることになる。

【０１７１】図５１は、第２実施形態におけるるスレッド間の遷移を示す図である。図５１におけるシーケンスは、図５０同様縦方向下向きが時間軸であり、４つの長方形を含む太枠で囲まれた四角形がスレッドとなる。太枠の中の一つの各四角形は一つの命令を実行するサイクルを表している。これらの四角形のうち、『』が記入されているものはWait_Until_Next命令を示す。またタスク(0)が各スレッドにおいて命令メモリ１００内のどの命令を実行するかを、四角形の右横の『PC』『PC+1』『PC+2』『PC+3』といったプログラムカウント値及びプログラムカウントからの相対値により表現している（尚本図は、タスク(0)の第１スレッドの第１命令の読み出し時におけるプログラムカウント値を基準にしている。）。

【０１７２】図４７においてステップＳ１～ステップＳ４の手順が行われることにより、タスク(0)の第１スレッドの第１命令が読み出される。読み出された命令の解読結果の判定がステップＳ１２において行われるが、タスク(0)の第１スレッド－第１命令はWait_Until_Next命令ではないのでステップＳ５に移行する。ステップＳ５への移行により、タスク(0)の第１スレッド－第１命令の実行が行われて、１命令分進行したことになる。

【０１７３】再度ステップＳ１～ステップＳ４の手順が行われることにより、タスク(0)の第１スレッドの第２命令が読み出される。読み出された命令の解読結果の判

定がステップＳ１２において行われるが、タスク(0)の第１スレッド－第２命令はWait_Until_Next命令であり、ステップＳ８への移行が行われる。ステップＳ８により変数iは１インクリメントされて『3』となり、ステップＳ９において読出先アドレス(adr+2)がタスク(0)の読み出し先アドレスとしてタスク別PC格納部110に格納される。

【０１７４】格納後、ステップＳ１０において変数kはインクリメントされ、ステップＳ２に移行する。このステップＳ１０における変数kのインクリメントにより、タスク(0)からタスク(1)への切り換えが、第１スレッドの第３命令,第４命令の実行を待たずに行われる。タスク(1)の第１スレッドの実行が四命令の実行により行われ、同様にタスク(2)、タスク(3)、タスク(4)、タスク(5)の実行が行われて、タスク(5)の第１スレッド－第４命令まで進んだとする。ステップＳ１０において変数kが決定され、ステップＳ２において変数iが『1』に初期化されて、ステップＳ３においてタスク(0)の読出先アドレスがタスク別PC格納部110から読み出される。ここで留意されたいのは、タスク(0)の第１スレッドの第２命令（Wait_Until_Next命令）の実行時に、PC+2（『PC』は第１命令のプログラムカウント値を示す。）がタスク(0)の読出先アドレスとしてタスク別PC格納部110に格納されている点である。このようにタスク別PC格納部110に(PC+2)が格納されているため、タスク(1)の第２スレッドの命令実行はPC+2から開始されることになる。

【０１７５】（第３実施形態における全体制御の説明）図４７は第３実施形態におけるI/Oプロセッサのフローチャートである。本図において、図４５、図４６の同一処理内容のステップには、図４５、図４６と同一参照符号を付してその説明を省略する。また図中の『変数Total』とは、タスク管理レジスタ１３の内容に基づき、セレクタ５５が出力する数値を示し、命令実行数の上限を指定するために用いられる。変数Totalはその初期状態において、スレッドにおける命令総数である『４』に初期化されるが、『２』にも更新され得る。

【０１７６】図５２は、第３実施形態のフローチャートによるスレッド間の遷移を示す図である。以降の説明では図５２において、タスク(0)の第１スレッドの第１命令、第２命令が実行済みであり、これから第３命令が読み出されようとしている。ステップＳ９３は、上記変数kをゼロクリアし、変数Totalを４に設定する。ステップＳ２は、上記変数iをゼロクリアする。ステップＳ３において、ゼロクリアされた変数kの読出先アドレス、即ち、タスク(0)の読出先アドレスをタスク別PC格納部110から読み出す。ステップＳ４において命令メモリ１００において読み出されたタスクの読出先アドレスを絶対アドレスとし、変数iをオフセット値としたアドレスから命令を取り出す。

【０１７７】ステップＳ８１において取り出された命令がCmp_And_Wait命令であるかを判定する。ステップＳ８２では命令解読制御部８１、演算実行部８４はCmp_And_Wait命令の第１オペランドに記述された状態監視レジスタjの指定と、第２オペランドに記述された即値とを参照してCmp_And_Wait命令が提示する事象の成立の真偽を判定する。状態監視レジスタjの保持値と即値とが一致せず、事象不成立が判定されると、ステップＳ８３に移行して、タスクkをwait状態に設定しステップＳ８４に移行する。ステップＳ８４では、プログラムカウントを進行させないよう、フリップフロップ２７及びセレクタ２５に読出先アドレスをCmp_And_Wait命令のアドレスに設定させてステップＳ８７に移行する。

【０１７８】ステップＳ８７は、図４６のフローチャートのステップＳ６同様、上限チェックを行うステップである。ステップＳ６における『変数i＝４』の判定は、命令の実行の４回の繰り返しをタスク切り換えの条件に課していた。これに対してステップＳ８７における『変数i＝変数Total』の判定は、命令の実行の繰り返し回数が変数Totalと等しいことをタスク切り換えの条件に課している。ここでの変数Totalは未更新であるから、『変数i＝変数Total』は実質『変数i＝４』の判定となる。Cmp_And_Wait命令はタスク(0)の第１スレッドにおいて3命令目に含まれており、一方変数i=2なので、ステップＳ８７からステップＳ７への移行が行われる。ステップＳ７において変数iがインクリメントされると、ステップＳ４における命令読み出しにより、Cmp_And_Wait命令の次順の命令が命令解読制御部８１に読み出される。読み出されるとステップＳ８１がNoとなりステップＳ８５では、命令解読制御部１１は直前の命令において状態不成立が判定され、読出先アドレスがCmp_And_Wait命令のアドレスに設定されたか（PC戻りが発生したか）を判定する。

【０１７９】ここでアドレス（adr+i）に格納されていた命令の直前命令とはCmp_And_Wait命令が相当し、このCmp_And_Wait命令の解読時にはPC戻りが発生している。そのためステップＳ８５はYesとなりステップＳ８６に移行する。ステップＳ８６では命令解読制御部１１は読み出されたアドレス（adr+i）の命令を廃棄するよう、汎用レジスタの格納値を無効化する旨の通知を行い、図５１のタスクシーケンスにおけるタスク(0)－第１スレッドの第４命令に示すように、アドレス（adr+i）に格納されていた命令を無効化してステップＳ８７に移行する。変数jの値は４であるからステップＳ８７において『変数i＝変数Total』の判定がなされるとYesとなり、ステップＳ８８に移行する。

【０１８０】ステップＳ８７がYesと判定されたことは、一スレッド分の命令実行が済んだことを意味するが、ステップＳ８８では、その一スレッドにおいてPC戻りが発生したかを判定する。このような判定を行うのはPC戻りが発生した場合はタスク別PC格納部110にその戻

り先を格納しておく必要があるからである。タスク(0)の場合は第４命令が廃棄され、第３命令においてPC戻りが発生しているので、ステップＳ８９においてフリップフロップ２７及びセレクタ２５に戻り先PC値を読出先アドレス(adr)として格納させてステップＳ１０に移行する。ステップＳ１０においてタスクラウンド管理部７３及びプライオリティエンコーダ７４に次のタスクを決定させた後、変数kをインクリメントしステップＳ９０に移行する。

【０１８１】ステップＳ９０～ステップＳ９２は、インクリメントされた変数kにより指示されるタスク、即ち次に実行されるべきタスクがwait状態にあるか否かを判定することにより、その次スレッドにおける命令数を切り換えるという重要な意味合いを持つ。順を追って説明するとステップＳ９０では、IDコンバータ５３はタスク管理レジスタ１３におけるタスク(k)に相当するビットを参照して、タスク(k)がwait状態にあるかを判定する。もしそうであればセレクタ５５はステップＳ９１において変数Totalに４を設定し、異なれば変数Totalに２を設定する。

【０１８２】タスク(1)は、wait状態ではないのでステップＳ９１において変数Totalが４に設定されてステップＳ２に移行する。ここでタスク(1)にはCmp_And_Wait命令が含まれていないものとすると、ステップＳ５における命令実行が四回分繰り返されて、次のタスクへの切り換えが行われる（以上の手順は、第１実施形態に記載されたものと同様であるから重複説明はあえて行わない。）。

【０１８３】Cmp_And_Wait命令を含んでいないタスク(2)、タスク(3)、タスク(4)についても、四命令の実行が順次行われて順々にタスクが切り換ってゆく。タスク(5)についての四命令の実行が済んでステップＳ１０に移行し、ステップＳ１０において変数kをゼロクリアして、ステップＳ９０に移行したとする。ステップＳ９０では、タスク管理レジスタ１３においてタスクkに割り当られたビットkが『1』であるか『0』であるかを判定する。この判定は、タスクkがwait状態であるか否かを見極めを意味する。ここで変数k＝０であり、タスク(0)は先程wait状態に設定されているから、ステップＳ９１において変数Totalに『２』を設定する。ここで留意すべきは、変数Totalは、スレッドに割り当る命令数の意味合いを持っている点である。これをステップＳ９１において『４』から『２』に更新することは、－スレッドにマッピングする命令数を『４』から『２』に削減することを意味する。そのため、タスク(0)の第２スレッドでは、変数iが1,2とインクリメントされている間だけ、つまりステップＳ８１～ステップＳ８５、ステップＳ５の処理の繰り返しが２回だけとなる。この際、タスク(0)の読出先アドレスとしてタスク別PC格納部110に格納されているのはCmp_And_Wait命令の読出先アドレスである。読出先アドレスのアドレスからCmp_And_Wait命令が読み出されると第１スレッド同様、ステップＳ８２において事象の成否の判定がなされる。この場合、事象は不成立で終わったものとすると、第１スレッドと同様の処理を行って読出先アドレスをCmp_And_Wait命令に戻して第２スレッドを終える。

【０１８４】Cmp_And_Wait命令を含んでいないタスク(1)、タスク(2)、タスク(3)、タスク(4)についても、四命令の実行が順次行われて順々にタスクが切り換ってゆき、タスク(5)についての四命令の実行が済んでタスク(0)の第３スレッドが始まろうとしている。ここでI/Oプロセッサにおける事象に変化が生じ、タスク(0)のCmp_And_Wait命令において指定されていた事象が成立していたとする。このためタスク(0)の第３スレッドでは、ステップＳ８２において命令解読制御部８１により事象成立が判定され、ステップＳ８０に移行して、タスク管理レジスタ１３におけるタスクkのwait状態が解除される。このようにwait状態が解除されると、ステップＳ８０からステップＳ８７への移行が行われ、ステップＳ８７において変数iと変数Totalとの比較が行われ、ステップＳ７において変数iのインクリメントがなされて、ステップＳ４において読出先アドレス(adr+1)から命令が読み出される。ステップＳ８１において読み出された命令がCmp_And_Wait命令であるかの判定が行われるが、Cmp_And_Wait命令ではないのでステップＳ８５へと移行する。ステップＳ８５では、直前の命令において状態不成立が判定されPC戻りが発生したかが判定される。Cmp_And_Wait命令により指定された事象が成立したので、ステップＳ８５がNoとなり、Cmp_And_Wait命令の次順命令は二度の事象成立待ちを経てやっとステップＳ５において実行される。ステップＳ５における次順命令の実行後、ステップＳ５からステップＳ８７への移行が行われ、変数iと変数Totalとの大小比較が行われる。但し第２スレッドでは変数Totalが２に設定されているので、タスク(0)の第３スレッドは２回実行されただけで終わってしまう。ステップＳ８７への移行後、ステップＳ８８においては、PC戻りの発生の真偽が判定されるがこの場合はNoであり、ステップＳ８９においてタスク(k)の読出先アドレス(adr+i-1)を読出先アドレス(adr)として格納してステップＳ１０に移行する。このようにしてタスク(0)の第３スレッドが済むと、Cmp_And_Wait命令を含んでいないタスク(1)、タスク(2)、タスク(3)、タスク(4)についても、四命令の実行が順次行われて順々にタスクが切り換ってゆく。タスク(5)についての四命令の実行が済んで変数kがゼロクリアされ、ステップＳ９０に移行したとする。

【０１８５】ステップＳ９０では、タスク管理レジスタ１３におけるタスク(k)に相当するビットを参照して、タスク(k)がwait状態にあるかを判定する。この場合タスク(0)のwait状態は解除されたので、ステップＳ９２

において変数Totalに４を設定してステップＳ２に移行する。このように変数Totalが４に戻ったので、タスク(0)の以降のスレッドは通常通りの四命令ずつ行われることになる。

【０１８６】（第４実施形態における全体制御の説明）図４８は、第４実施形態におけるI/Oプロセッサの全体制御を示すフローチャートである。図４８のフローチャートにおけるステップＳ１～ステップＳ９の手順は図４６のフローチャートと同様である。図４８において新規なのは、３０番台の参照符号が付与されたステップＳ３０、ステップＳ３１、ステップＳ３２である。ステップＳ３０は、緊急状態遷移許可信号がLow値であるか否かを判定し、High値ならばステップＳ３１に移行してステップＳ３１において緊急タスクを次に実行すべきタスクとして決定する。

【０１８７】逆に緊急状態遷移許可信号がLow値ならば、ステップＳ３２に移行してラウンド値taskneに基づいて、通常タスクの中から次に実行すべきタスクをプライオリティエンコーダ８０に決定させる。図５３は、第４実施形態におけるスレッド間の遷移を示す図である。本図は、通常状態が始まりにおいて、ステップＳ３１の処理がスキップされて、図５２のタスク(0)の第１スレッド、タスク(2)の第１スレッド、タスク(3)の第１スレッド、タスク(4)の第１スレッド、タスク(5)の第１スレッドが順次行われる。

【０１８８】taskneの実行が一巡したところで、緊急状態遷移許可信号のHigh期間が発生したとすると、ステップＳ２～ステップＳ９において一つのタスクが実行された後、ステップＳ３０からステップＳ３１への分岐が行われる。分岐後、cstate記憶部８５に示されたcstateのHigh値、Low値に応じて緊急タスクの実行を行う。

（第５実施形態における全体制御）図４９は第５実施形態におけるマルチタスク手順のフローチャートである。本図が第１実施形態におけるマルチタスク手順のフローチャートと異なるのは、ステップＳ４とステップＳ６との間にステップＳ４１～ステップＳ４４、Ｓ４６、Ｓ４７が挿入されている点である。

【０１８９】ステップＳ４１は、命令メモリ１００から読み出された命令がsleep命令であるかを判定する。もしsleep命令であれば、ステップＳ４２へと移行する。ステップＳ４２では、タスク管理レジスタ１３はタスク(k)に対応するビットを０に反転する。その後ステップＳ４３において変数iを1インクリメントして、ステップＳ４４においてタスク(k)の読出先アドレス(adr+i-1)を読出先アドレス(adr)として格納した後にステップＳ４５に移行する。移行後、ステップＳ４５では、インバータ９４及びOR回路９５はタスク管理レジスタ１３においてsleep状態が指定されたビット抜きのラウンド値を生成し、これをプライオリティエンコーダ７４に出力して、次に実行すべきタスク(k)を決定させる。ここで留意すべきは、先に述べたように『変数k』は命令メモリ１００に記憶されている６本のタスクのそれぞれを指示するための変数であり、これの決定はタスクからタスクへの切り換えを意味する。

【０１９０】ここでステップＳ４５がステップＳ１０と異なるのは、ステップＳ１０は変数iが４になった時点で実行されるのに対して、ステップＳ４５は変数iの値に拘らず、sleep命令が解読された時点で実行される点である。このようにステップＳ４５が変数iの値に拘らずsleep命令の解読後に実行されるため、タスクからタスクへの切り換えは、四回の命令を待たずして行われることになる。

【０１９１】ステップＳ４６では、読出先アドレス(adr+1)から読み出された命令がタスク管理レジスタ内の何れかのビットに対するwake命令であるかを判定する。もし異なれば、ステップＳ５において読み出された命令を解読して実行するがもしそうであれば、ステップＳ４７において当該wake命令において指定されたjビットを操作することにより、タスクjのsleep状態を解除してステップＳ６に移行する。

【０１９２】図５４は、第５実施形態におけるスレッド間の遷移を示す図である。本図においてタスク(1)の第１スレッドにおいて印を付しているのは、タスク(1)の第１スレッドの第２命令がsleep_task命令であることを示している。sleep_task命令の実行時には、第２実施形態におけるCmp_And_Wait命令の実行時と同様、第２命令の解読においてステップＳ４１～ステップＳ４４が実行されてタスク(1)がsleep状態に設定された後にタスク切り換えがなされ、第３命令及び第４命令は事実上廃棄されることになる。

【０１９３】第１実施形態に述べたようなタスク間の切り換えが順次行われて、タスク(2)、タスク(3)、タスク(4)、タスク(5)というようにタスクの巡回が行われてステップＳ４５に移行したとする。ステップＳ４５においてプライオリティエンコーダ７４は、sleep状態にあるタスクのビットがマスクされたラウンド値taskneにより、次に実行すべきタスクを決定する。ここでタスク管理レジスタ１３においてタスク(1)に割り当てられているビットは、タスク(1)－第１スレッドの第２命令においてsleep状態に設定されている。そのためタスク(1)の実行はスキップされ、タスク(2)の第２スレッドが続いて実行される。

【０１９４】同様のタスク間の切り換えが順次行われて、タスク(2)、タスク(3)、タスク(4)、タスク(5)というようにタスクの巡回が行われてステップＳ４５に移行したとする。この巡回において、タスク(5)の第２スレッドの第３命令がタスク(1)のsleep状態解除を指定したwake_task命令であり、命令メモリ１００から読み出されたものとする。wake_task命令が命令メモリ１００から読み出されると、ステップＳ４６の判定においてYes

となり、ステップＳ４７に移行する。ステップＳ４７では、タスク管理レジスタ１３のタスク(1)に割り当られたビットをオフにすることにより、タスク(1)のsleep状態の解除を行う。解除後、残りの命令をも同様に実行して、第２スレッド分の処理を終える。

【０１９５】タスク(5)の終了後、タスクがタスク(0)への切り換えられたものとする。またタスク(0)の第３スレッドが実行されて、ステップＳ４５へと移行したとする。ステップＳ４５では、タスク管理レジスタ１３においてタスク(1)に割り当られているビットがオフであるかの判定がなされる。タスク(5)の第２スレッドにおいて、タスク(1)の割り当てビットがオンに設定されたのは先に述べた通りであるから、ステップＳ４５からステップＳ２、Ｓ３への移行が行われて、タスク(1)の実行が再開される。

【０１９６】

【発明の効果】以上の本発明によれば、n個のタスクを実行対象タスクとしたプロセッサであって、n個のタスク識別子を出力し、そのタスクに割り当てられた数の命令が実行されると次のタスク識別子を所定の順序で出力する実行タスク指示手段と、出力されるタスク識別子で特定されるタスク中の実行すべき命令を順次指定する命令指定手段と、指定された命令を実行する実行手段とを備えているので、本プロセッサは割込信号の発行を待つのでは無く、オーディオアウトタスク、ビデオアウトタスクを含めたn個のタスクを所定の命令数ずつ順次実行する。

【０１９７】このように順次実行されるので、オーディオアウトタスク、ビデオアウトタスクの処理を完遂させるための動作クロック数の最適な下限値を、所定の周期において実行せねばならない命令の最低消化数と、実行周期と、タスク総数とから一義的に導出することができ、それらの最適な下限値から全タスクを動作させるための動作サイクル数の最適な下限値を一義的に計算することができる。

【０１９８】動作クロック数の最適な下限値が一義的に計算されるため、動作クロック信号を一段と高いものに定めずとも、オーディオアウトタスク、ビデオアウトタスクのリアルタイム性を保証でき、動作クロックが低速なタイプが多い民生機器に搭載される場合でも、MPEGストリームを好適に復号することができる。ここで実行タスク指示手段は、実行された命令をカウントし、カウント値が実行中タスクに割り当てられた命令数になるとタスク切替信号を発生するタスク切替信号発生器と、タスク切替信号が発される度に次順位のタスク識別子を生成して出力するタスク識別子出力部とにより構成することができる。

【０１９９】ここでタスク識別子出力部は各タスク識別子を何番目に出力するかを示す出力順位を、n個の全てのタスクについて記憶する順位記憶部と、タスク切替信号が発される度に、順位記憶部において出力順位が次順位となるタスク識別子を命令指定手段に出力する選択出力部とにより構成することができる。ここでタスク識別子出力部は、各タスク識別子を何番目に出力するかを示す出力順位を、n個の全てのタスクについて記憶する順位記憶部と、タスク切替信号が発される度に、順位記憶部において出力順位が次順位となるタスク識別子を命令指定手段に出力する選択出力部とにより構成することができる。

【０２００】ここで何れかのタスクは緊急扱いを要する旨の緊急宣言命令を含み、前記実行タスク指示手段は、プロセッサ外部から入力される所定信号を監視することにより、当該プロセッサ外部が緊急期間内であるか、緊急期間外であるかを判定する監視手段と、所定信号が緊急期間内である場合に、緊急扱いすべきタスクである緊急タスクの識別子を保持する緊急タスクレジスタと、緊急期間外においてタスク切替信号が発されれば、選択出力部が未出力のタスクから選んで出力したタスク識別子を実行手段にスルー出力し、緊急期間内においとてタスク切替信号が発されれば、タスク切替信号のm回の発信につき(m≧2)一回の比率にて、緊急タスクレジスタが保持しているタスク識別子を実行手段に出力する出力比率制御部とを構成することができる。

【０２０１】本構成によれば映像出力に関する非同期イベントタスクの識別子を緊急タスクレジスタに保持させておき、出力比率制御部が緊急期間内にタスク切替信号のm回の発信につき(一回の比率にて、緊急タスクレジスタが保持しているタスク識別子を出力すれば、ディスプレィにおける水平ブランキング期間、垂直ブランキング期間内にビットストリームを実行させておくことができる。これにより映像信号への伸長を表示期間に間に合わせることができる。このように再生系ハードウェアの処理が周期的にオフになる期間に同期して、効率的に動画データを処理すれば動画データタスクのリアルタイム性を向上させることができる。

【０２０２】ここで何れかのタスクは自身の実行を休眠状態に設定すべき旨の自発休眠命令を含み、前記実行手段は更に、命令指定手段が次に実行すべき命令として指定した命令を解読する解読手段を備え、前記実行タスク指示手段は更に、解読手段による解読結果が自発休眠命令であれば、当該命令を含むタスクを休眠扱いを要するタスクとしてそのタスク識別子を保持する休眠タスクレジスタを備え、前記選択出力部は、順位記憶部が記憶における次順位のタスク識別子が、休眠タスクのタスク識別子なら、その次順位のタスク識別子を出力するように構成することができる。

【０２０３】MPEGストリームの復号処理においては、処理が必要となる周期が大きく異なるタスクを並列実行せねばならない。例えば、マクロブロックに含まれている輝度ブロック、色差ブロックをバッファ間で転送させる

ための転送制御用のタスクはマクロブロックにおいて、少なくとも６回以上の頻度にて起動せねばならない。これに対して、MPEGストリームからエレメンタリィストリームを抽出するタスクは少ない頻度で起動すればよい。

【０２０４】休眠タスクレジスタは、タスクに自発休眠命令が含まれていれば、当該命令を含むタスクを休眠扱いを要するタスクとしてそのタスク識別子を保持し、前記選択出力部は、順位記憶部が記憶における次順位のタスク識別子が、休眠タスクのタスク識別子なら、その次順位のタスク識別子を出力するので、実行頻度が少ないタスク実行を省くことができる。これにより非同期イベント処理の起動頻度の差違を考慮して、各非同期イベントタスクに自ら休眠状態に入る機会を与えて、タスクの巡回効率を向上させることができる。

【０２０５】また、n個のタスクを実行対象タスクとしたプロセッサであって、決められたサイクル数でタスクを順次選択するタスク選択手段と、タスクと一対一の関係でｎ個の命令指定情報を有し、タスク選択手段によってタスクが選択されるとそれに対応した命令指定情報を有効にすると共に、その情報を起点として、次に読み出すべき命令を指定する情報を動的に生成する命令指定手段と、命令指定手段にて指定された命令を読み出し実行する実行手段と構成することができる。

【０２０６】ここで命令指定手段は、n個のタスクと一対一に対応し、対応するタスクに関して次に読み出すべきアドレス値を命令指定情報として保持しているｎ個とのアドレスレジスタと、タスク選択手段が選択したタスクに対応するアドレスレジスタを選択し、そのアドレス値を出力させるレジスタ選択部と、レジスタ選択部によりアドレスレジスタが選択されると、当該アドレスレジスタが保持しているアドレス値をカウント初期値として保持するカウント値レジスタと、カウント値レジスタが保持しているカウント値を、サイクル毎に、インクリメントするインクリメンタと、インクリメントされたカウント値を前記次に読み出すべき命令を指定する情報の更新値として保持する読出先アドレス保持部とにより構成することができる。

【０２０７】ここで前記命令指定手段は、インクリメンタがインクリメントしたカウント値をスルー出力して読出先アドレス保持部に保持させると共に、タスク選択手段がタスクを選択すると、当該タスクに対応するアドレスレジスタのアドレス値を選択出力して読出先アドレス保持部に保持させる第１セレクタと、次順のタスクに切り換わる際に、カウント値レジスタが保持しているカウント値を用いて、切り換え直前まで選択部で選択されていたアドレスレジスタが格納するアドレス値を書き換える第１書換部とにより構成することができる。

【０２０８】前記タスク切り換えを指定する旨の自発切換命令が含まれており、前記実行手段は読出先アドレス保持部が保持しているアドレスにて特定される命令を解読する解読手段を備え、命令指定手段は更に解読手段による解読結果が自発切換命令であれば、読出先アドレス保持部が保持している命令指定情報を用いて、切り換え直前まで選択部で選択されていたアドレスレジスタの格納するアドレス値を書き換える第２書換部を備え、前記第１セレクタは、解読手段による解読結果が自発切換命令であれば、当該次のタスクに対応するアドレスレジスタのアドレス値を選択出力して読出先アドレス保持部に保持させるように構成することができる。

【０２０９】非同期イベント処理には、内部バッファとSDRAMとの間の入出力を制御するものが多くあるが、SDRAMの状態は外的要因によって変動することが多い。そのためメモリベリファイや、あるメモリから値を読み出して、この値を用いた演算を行い、その結果を同一メモリに書き戻すという同一メモリをアクセス先としたメモリアクセス動作は、一個のスレッドにおいて行うのが望ましい。逆に、メモリから値を読み出す読出命令と、この値を用いた演算の演算結果を同一メモリに書き戻す書込命令とが別々のスレッドに配置されてしまうと、１つ目のスレッドが巡ってきてから、２つ目のスレッドが巡ってくる迄の間にメモリ状態が変動してしまい、読出命令は完遂し得たものの、書込命令が完遂できない場合が生じる。

【０２１０】しかし上記の構成では、自発切換命令を読出命令、書込命令の直前に配置しておけば、第２切り換部がアドレスレジスタの内容を自発切換命令が含まれていたアドレスの次のアドレスに書き換えると共に、第１セレクタが当該次のタスクに対応するアドレスレジスタのアドレス値を選択出力して読出先アドレス保持部に保持させるので、自発切換命令の直後で本タスクの命令実行を打ち切ることができる。即ち、同じタスクのタスク識別子が出力された際、当該タスクは自発切換命令の直後に配置された読出命令、書込命令を、同一のスレッド内で実行することができる。

【０２１１】これにより、タスク切り換えによる中断を望まない四命令を１タスクに割り当てる等、どの所定数の命令を１スレッドに割り当てるかを自発切換命令の配置により、独自に決めておくことができる。また一部タスクの消化を四命令の実行を待たずに打ち切ることにより他のタスクを早めに消化することもできる。

【図面の簡単な説明】

【図１】AVデコーダの内部構成を示す図である。

【図２】MPEGストリームの階層構造と、AVデコーダの構成要素間の動作タイミングとを示すタイミングチャートである。

【図３】第１実施形態におけるI/Oプロセッサの内部構成を示す図である。

【図４】第１実施形態における命令読出回路１０の内部構成を示す図である。

【図５】命令読出回路１０内のセレクタ２５の出力論理

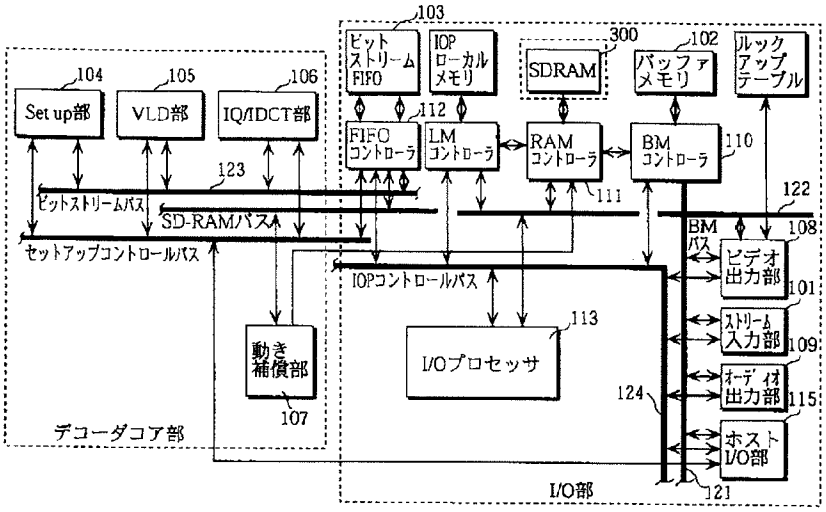
表を示す図である。
【図６】第１実施形態におけるタスク管理部１５の内部構成を示す図である。
【図７】（ａ）タスク識別子taskidのビット構成を示す図である。
（ｂ）ラウンド値のビット構成を示す図である。
【図８】第１実施形態において命令メモリ１００に記憶されている非同期イベントタスクがどのように構成されているかを示す図である。
【図９】レジスタをオペランドに指定した算術演算命令の命令フォーマットを示す。
【図１０】レジスタにて読出先アドレス、書込先アドレスを指定したメモリロード命令／メモリストア命令の命令フォーマットを示す。
【図１１】8bit、11bitの即値の指定が可能な比較命令／演算命令／分岐命令の命令フォーマットを示す。
【図１２】第１実施形態において、IPC,IF1,IF2,DECPCがどのように更新されるかを示すタイミングチャートである。
【図１３】タスク(0)からタスク(5)までにどのようにスレッドを割り当てるかを示す図である。
【図１４】第２実施形態において命令メモリ１００に記憶されている非同期イベントタスクがどのような命令にて構成されているかを示す図である。
【図１５】wait_until_next命令の命令フォーマットの一例を示す。
【図１６】第２実施形態において、IPC,IF1,IF2,DECPCがどのように更新されるかを示すタイミングチャートである。
【図１７】第２実施形態においてタスク(0)からタスク(5)までにどのようにスレッドを割り当てるかを示す図である。
【図１８】第３実施形態におけるAVデコーダの内部構成を示す図である。
【図１９】第３実施形態における命令読出回路１０の内部構成を示す図である。
【図２０】第３実施形態における命令読出回路１０内のセレクタ２５の出力論理表を示す図である。
【図２１】第３実施形態におけるスレッドマネージャー６１の内部構成を示す図である。
【図２２】タスク管理レジスタ１３においてwait状態管理用に割り当られた6bitのビット構成を示す図である。
【図２３】第３実施形態における状態監視レジスタのビット構成を示す図である。
【図２４】（ａ）第３実施形態においてどのタスクにCmp_and_Wait命令が含まれているかを示す図である。
（ｂ）事象j0,j3,j4,j5が不成立である状態において、図２４（ａ）に示す各タスクにどのようにスレッドが割り当てられるかを示す図である。
（ｃ）事象j0のみが成立したフレームにおいて、図２４（ａ）に示す各タスクにどのようにスレッドが割り当てられるかを示す図である。
（ｄ）事象j0が成立したフレーム以降において、図２４（ａ）に示す各タスクにどのようにスレッドが割り当てられるかを示す図である。
【図２５】Cmp_and_Wait命令の命令フォーマットの一例を示す。
【図２６】Cmp_and_Wait命令解読時においてIPC,IF1,IF2,DECPCがどのように更新され、タスク管理レジスタ１３のビット及びカウンタの上限値がどう設定されるかを示すタイミングチャートである。
【図２７】Cmp_and_Wait命令が待っていた事象の未成立時のフレームにおいて、IPC,IF1,IF2,DECPCがどのように更新され、タスク管理レジスタ１３のビット及びカウンタの上限値がどう設定されるかを示すタイミングチャートである。
【図２８】Cmp_and_Wait命令が待っていた事象の成立時のフレームにおいて、IPC,IF1,IF2,DECPCがどのように更新され、タスク管理レジスタ１３のビット及びカウンタの上限値がどう設定されるかを示すタイミングチャートである。
【図２９】事象の成立時以降のフレームにおいて、IPC,IF1,IF2,DECPCがどのように更新され、タスク管理レジスタ１３のビット及びカウンタの上限値がどう設定されるかを示すタイミングチャートである。
【図３０】第３実施形態におけるタスク管理部１５の内部構成を示す図である。
【図３１】第３実施形態におけるスケジューラ６２の内部構成を示す図である。
【図３２】セレクタ８２が緊急タスクの識別子を出力する条件を示す図である。
【図３３】cstate記憶部８５が記憶しているモードの状態遷移図である。
【図３４】緊急状態遷移許可信号、タスク切替信号chg_task_ex、cstateの遷移に応じて、どのタスク識別子が出力されるかのタイミングを示すタイミングチャートである。
【図３５】（ａ）通常状態において各タスクにどのようにスレッドを割り当てるかを示す図である。
（ｂ）緊急状態において各タスクにどのようにスレッドを割り当てるかを示す図である。
【図３６】第４実施形態におけるタスク管理部１５の内部構成を示す図である。
【図３７】タスク管理レジスタ１３においてsleep状態に割り当られたビットの割り当てを示す図である。
【図３８】第４実施形態におけるスケジューラ６２の内部構成を示す図である。
【図３９】第４実施形態において、sleep命令及びwake命令が各タスクにどのように配置されているかを示す図である。

【図４０】sleep命令及びwake命令の命令フォーマットの一例を示す。
【図４１】第４実施形態においてIPC,IF1,IF2,DECPCがどのように更新されるかを示すタイミングチャートである。
【図４２】（ａ）～（ｄ）第４実施形態において、各タスクにどのようにスレッドを割り当てるかを示す図である。
【図４３】第５実施形態におけるタスク管理部１５の内部構成を示す図である。
【図４４】第５実施形態におけるスケジューラ６２の内部構成を示す図である。
【図４５】第１実施形態におけるI/Oプロセッサの全体制御を示すフローチャートである。
【図４６】第２実施形態におけるI/Oプロセッサの全体制御を示すフローチャートである。
【図４７】第３実施形態におけるI/Oプロセッサの全体制御を示すフローチャートである。
【図４８】第４実施形態におけるI/Oプロセッサの全体制御を示すフローチャートである。
【図４９】第５実施形態におけるI/Oプロセッサの全体制御を示すフローチャートである。
【図５０】第１実施形態におけるスレッドの割り当てがどのように進行するかを示す図である。
【図５１】第２実施形態におけるスレッドの割り当てがどのように進行するかを示す図である。
【図５２】第３実施形態におけるスレッドの割り当てがどのように進行するかを示す図である。
【図５３】第４実施形態におけるスレッドの割り当てがどのように進行するかを示す図である。
【図５４】第５実施形態におけるスレッドの割り当てがどのように進行するかを示す図である。
【図５５】第１実施形態におけるスレッド割り当てにてビデオアウトタスク、オーディオアウトタスクを実行する場合に動画ストリーム、オーディオストリームがどのように処理されるかを示すタイミングチャートである。
【図５６】タスク別にサイクル数を可変にする場合のスレッドマネージャーの構成図である。
【図５７】割込信号にてオーディオアウトタスク、ビデオアウトタスクを実行する場合に動画ストリーム、オーディオストリームの処理を何時までに完遂せねばならないかを示すタイミングチャートである。

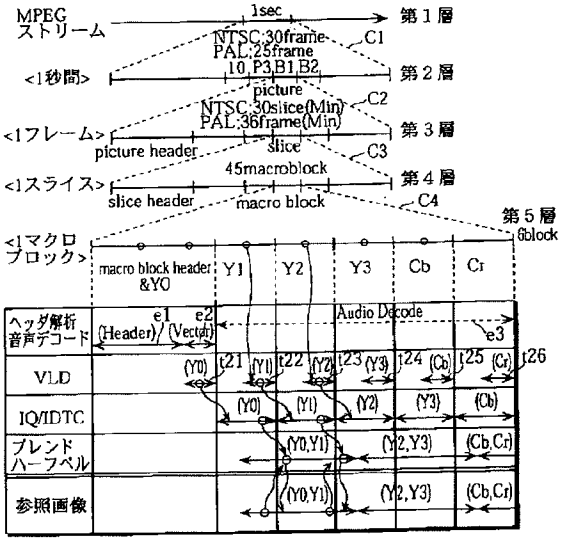
【符号の説明】

１０　命令読出回路
１１　命令解読制御部
１２　レジスタセット
１３　タスク管理レジスタ
１４　演算実行部
１５　タスク管理部
２０　IF1+1保持部
２１　increment回路
２２　IF2保持部
２３　DECPC保持部
２４　タスク別PC格納部
２５　4入力-1出力セレクタ
２６　2入力-1出力セレクタ
５２　カウンタ
５３　コンバータ
５４　比較器
５５　セレクタ
６１　スロットマネージャー
６２　スケジューラ
７１　タスクID保持部
７２　タスクID保持部
７３　タスクラウンド管理部
７４　プライオリティエンコーダ
７５　タスクラウンド管理部
７６　コンバータ
７７　緊急タスクマスク部
８０　プライオリティエンコーダ
８１　命令解読制御部
８２　セレクタ
８３　タスクID保持部
８４　演算実行部
８５　CSTATE記憶部
９３　タスクスキップ管理回路
９４　インバータ
１００　命令メモリ
１０１　ストリーム入力部
１０２　バッファメモリ
１０４　Setup部
１０５　VLD部
１０６　IQ/IDCT部
１０７　動き補償部
１０８　ビデオ出力部
１０９　オーディオ出力部
１１０　BMバッファメモリコントローラ
１１１　RAMコントローラ
１１２　FIFOコントローラ
１１３　プロセッサ
１１５　ホストI/O部
１２１　バッファメモリバス
１２２　SD-RAMバス
１２３　ビットストリームバス
１２４　コントロールバス

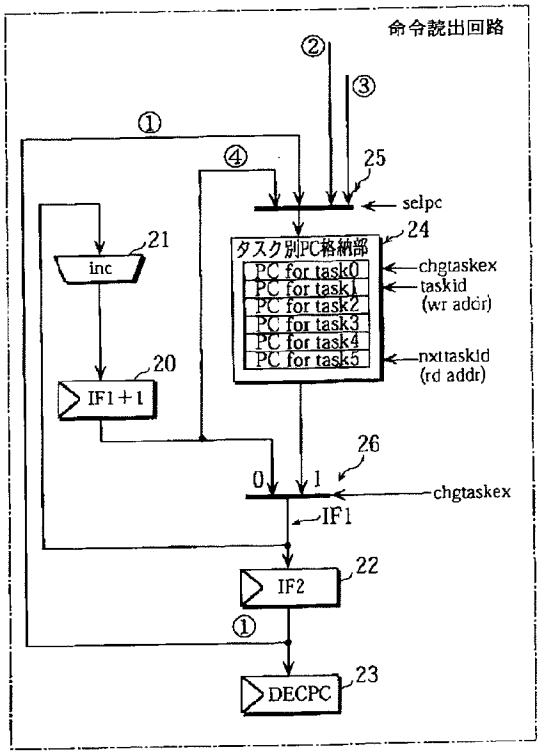
【図１】



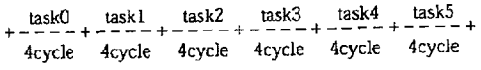
【図２】



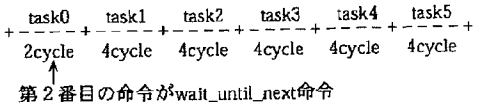
【図４】



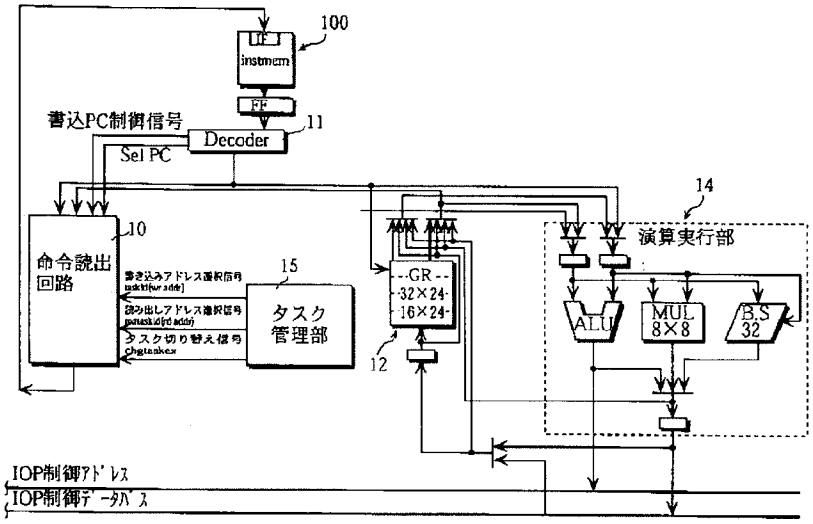
【図１３】



【図１７】



【図３】

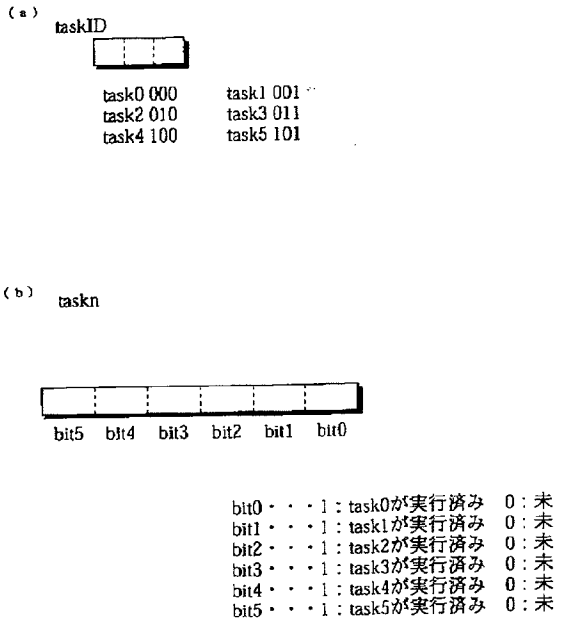


【図５】

セレクタ25の出力論理

wait_until_next命令の解読時	セレクタ26から出力されたアドレス(ルート①)
絶対アドレス型分岐命令の解読時	デコーダ11から出力されたアドレス(ルート②)
間接アドレス型分岐命令の解読時	ALU14から出力されたアドレス(ルート③)
chg_task_id信号の出力時	IF2から出力されたアドレス(ルート④)

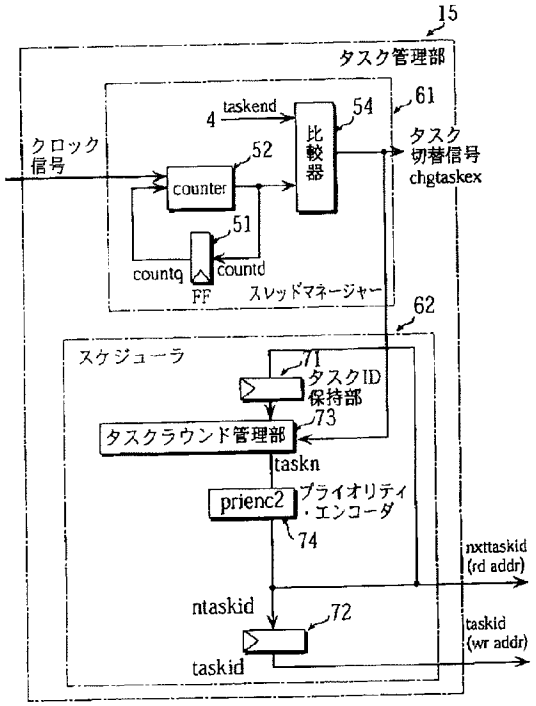
【図７】



【図８】

task0	task1	task2	task3	task4	task5
命令0-0	命令1-0	命令2-0	命令3-0	命令4-0	命令5-0
命令0-1	命令1-1	命令2-1	命令3-1	命令4-1	命令5-1
命令0-2	命令1-2	命令2-2	命令3-2	命令4-2	命令5-2
命令0-3	命令1-3	命令2-3	命令3-3	命令4-3	命令5-3
命令0-4	命令1-4	命令2-4	命令3-4	命令4-4	命令5-4
命令0-5	命令1-5	命令2-5	命令3-5	命令4-5	命令5-5
命令0-6	命令1-6	命令2-6	命令3-6	命令4-6	命令5-6
命令0-7	命令1-7	命令2-7	命令3-7	命令4-7	命令5-7
命令0-8	命令1-8	命令2-8	命令3-8	命令4-8	命令5-8
命令0-9	命令1-9	命令2-9	命令3-9	命令4-9	命令5-9
命令0-10	命令1-10	命令2-10	命令3-10	命令4-10	命令5-10
命令0-11	命令1-11	命令2-11	命令3-11	命令4-11	命令5-11
命令0-12	命令1-12	命令2-12	命令3-12	命令4-12	命令5-12
命令0-13	命令1-13	命令2-13	命令3-13	命令4-13	命令5-13
命令0-14	命令1-14	命令2-14	命令3-14	命令4-14	命令5-14
命令0-15	命令1-15	命令2-15	命令3-15	命令4-15	命令5-15

【図６】

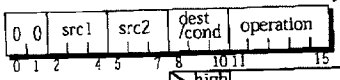


【図１４】

task0	task1	task2	task3	task4	task5
命令0-0	命令1-0	命令2-0	命令3-0	命令4-0	命令5-0
命令0-1 ●wun 命令	命令1-1	命令2-1	命令3-1	命令4-1	命令5-1
命令0-2 LD mem1,r0	命令1-2	命令2-2	命令3-2	命令4-2	命令5-2
命令0-3 LD buf1,r1	命令1-3	命令2-3	命令3-3	命令4-3	命令5-3
命令0-4 mul r0,r1,r2	命令1-4	命令2-4	命令3-4	命令4-4	命令5-4
命令0-5 ST r2,mem1	命令1-5	命令2-5	命令3-5	命令4-5	命令5-5
命令0-6	命令1-6	命令2-6	命令3-6	命令4-6	命令5-6
命令0-7	命令1-7	命令2-7	命令3-7	命令4-7	命令5-7
命令0-8	命令1-8	命令2-8	命令3-8	命令4-8	命令5-8
命令0-9	命令1-9	命令2-9	命令3-9	命令4-9	命令5-9

【図９】

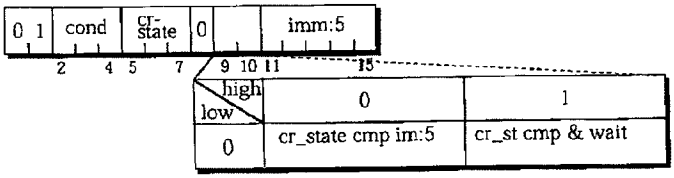
レジスタ指定型算術演算命令



operation

high / low	00	01	10	11
000	src1+src2=dst ADD	src1-src2=dst SUB	src1×src2=dst MUL	src1 cmp src2(cond) CMP
001	src1≪src2=dst	src1≫src2=dst	src1≫src2=dst	src1≫(32-src2)=dst
010	src1 and src2=dst AND	src1 or src2=dst OR	src1 xor src2=dst XOR	src1 test src2(cond) TST
011	src1 andn src2=dst ANDN	src1 orn src2=dst ORN	src1 xnor src2=dst XNOR	
100	src1=dst MOV	-src1=dst NOT		
101		Mask G(src1)=dst MASKG	NOP NOP	bra src1,pc+1=dst BRA
110	BIN0	BIN1	BIN2	BIN3
111	BEX0	BEX1	BEX2	BEX3

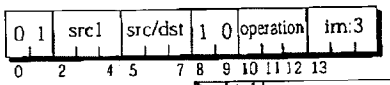
【図２５】



high / low	0	1
0	cr_state cmp im:5	cr_st cmp & wait

【図１０】

メモリロード/メモリストア命令



operation:

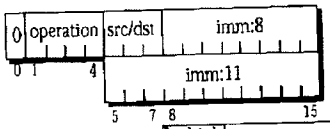
high / low	0	1
10	Mem(im:3+src1) ⇒dst_REG	src_REG⇒ Mem(im:3+src1)

【図３２】

iexecmode	!cstate	セレクタ52の出力
1	0	taskIde
0	1	taskidne
0	0	taskIdne
1	1	taskidne

【図１１】

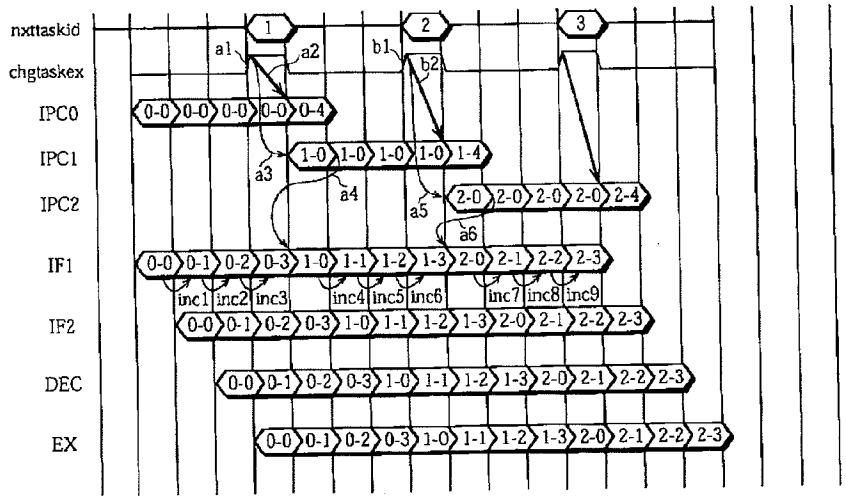
即値付き比較命令/即値付き演算命令/絶対アドレス指定分岐命令



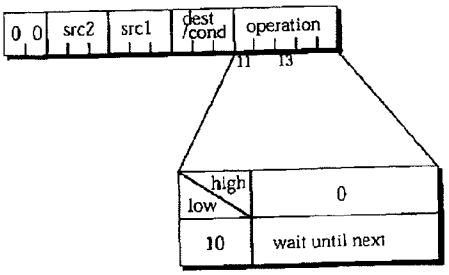
operation:

high / low	00	01	10	11
00	src+im8⇒dst ADD	src-im8=dst SUB	src cmp im8(==)	srccmp im8(!=)
01	src or im8⇒dst OR	src andn im8=dst AND	src cmp im8(>=)	src cmp im8(<)
10	src≪ im8⇒dst LSL	im8≪24=dst	s×im8⇒d MUL	src test im8(!=)
11	im:8⇒dst MOV	(s≪8)+ im8⇒dst	im:11⇒r0 MOV	bra im:11 BRAI

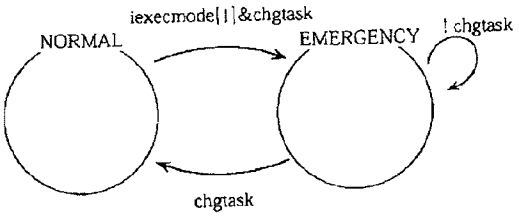
【図１２】



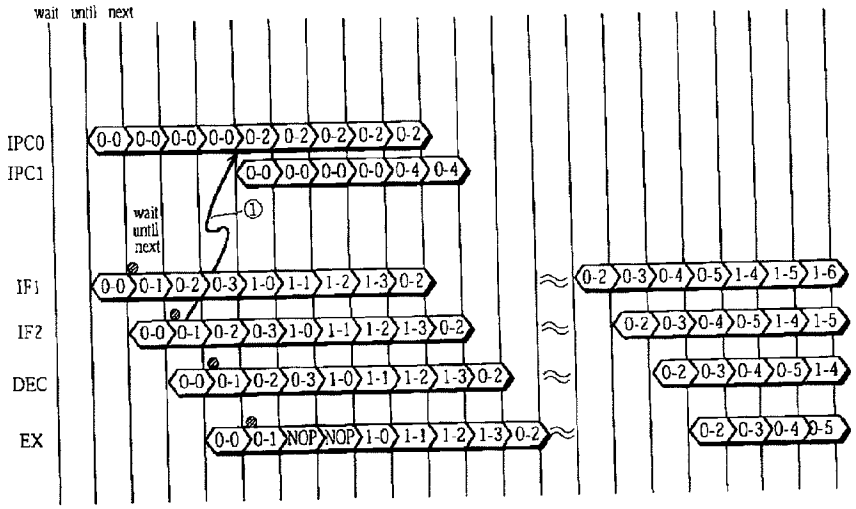
【図１５】



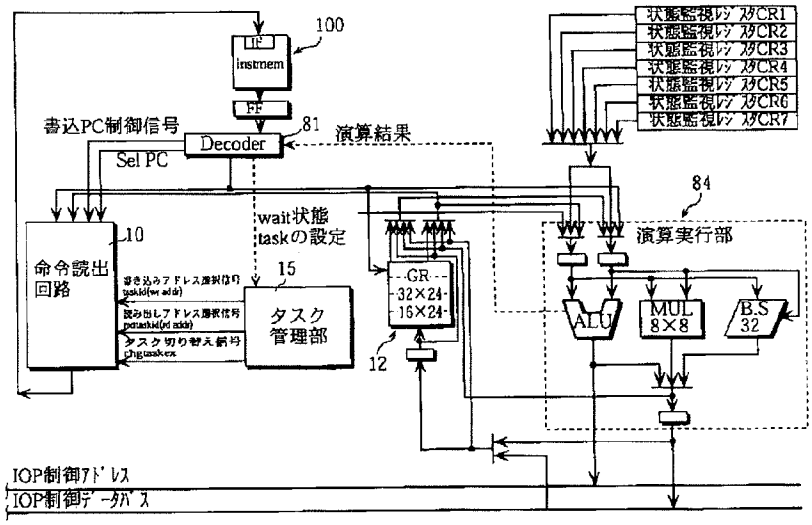
【図３３】



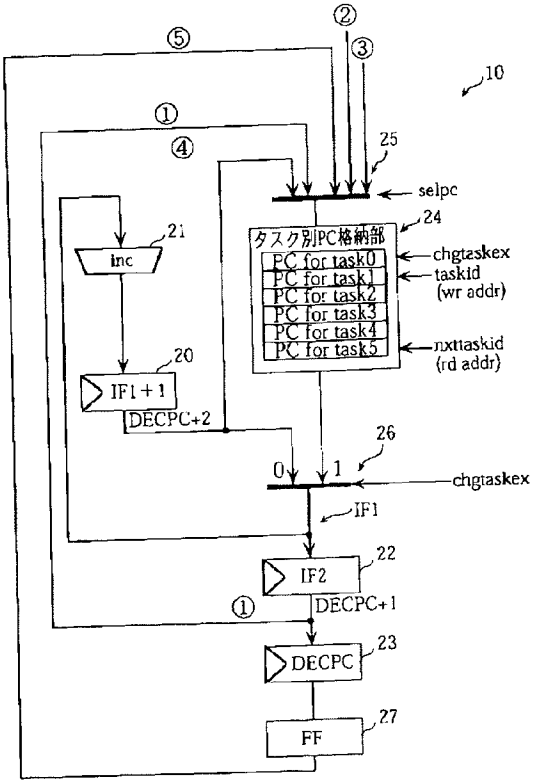
【図１６】



【図１８】

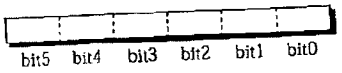


【図１９】



【図２２】

waitビット



bit0・・・1：task0を wait 状態に設定　0：未
bit1・・・1：task1を wait 状態に設定　0：未
bit2・・・1：task2を wait 状態に設定　0：未
bit3・・・1：task3を wait 状態に設定　0：未
bit4・・・1：task4を wait 状態に設定　0：未
bit5・・・1：task5を wait 状態に設定　0：未

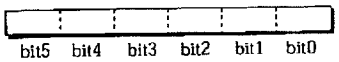
【図２０】

セレクタ25の出力論理

wait_until_next命令の解読時	セレクタ26から出力されたアドレス(ルート①)
絶対アドレス型分岐命令の解読時	デコーダ11から出力されたアドレス(ルート②)
間接アドレス型分岐命令の解読時	ALU14から出力されたアドレス(ルート③)
chg_task_id信号の出力時	IF2から出力されたアドレス(ルート④)
cmp_and_wait命令の解読時	FF27から出力されたアドレス(ルート⑤)

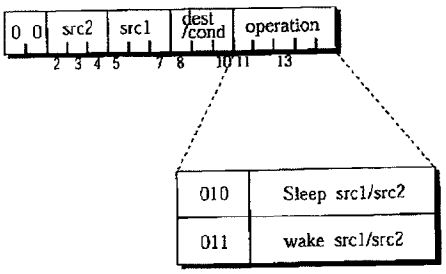
【図３７】

sleep ビット

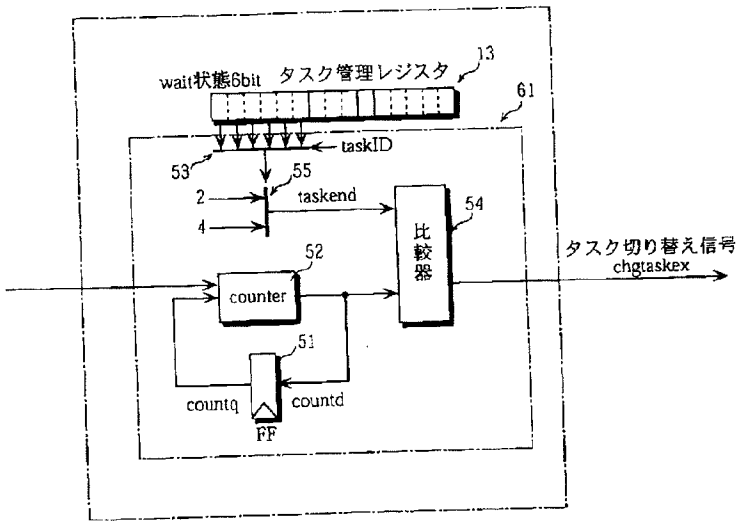


bit0・・・0：task0を sleep 状態に設定　1：未
bit1・・・0：task1を sleep 状態に設定　1：未
bit2・・・0：task2を sleep 状態に設定　1：未
bit3・・・0：task3を sleep 状態に設定　1：未
bit4・・・0：task4を sleep 状態に設定　1：未
bit5・・・0：task5を sleep 状態に設定　1：未

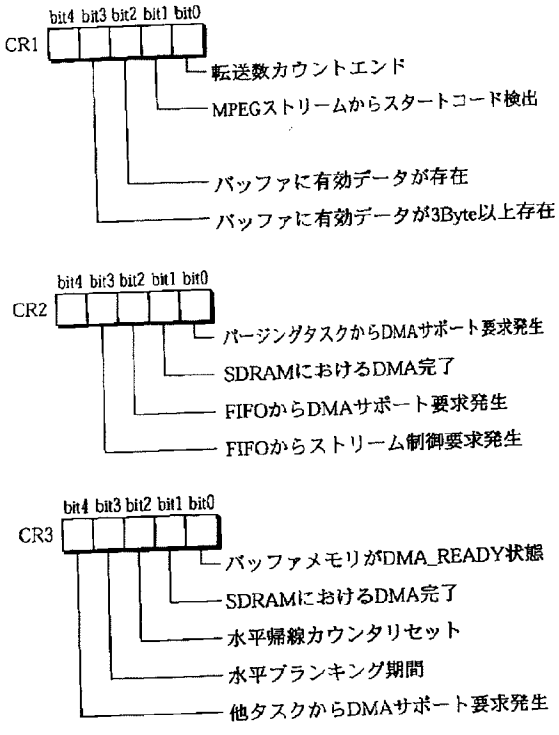
【図４０】



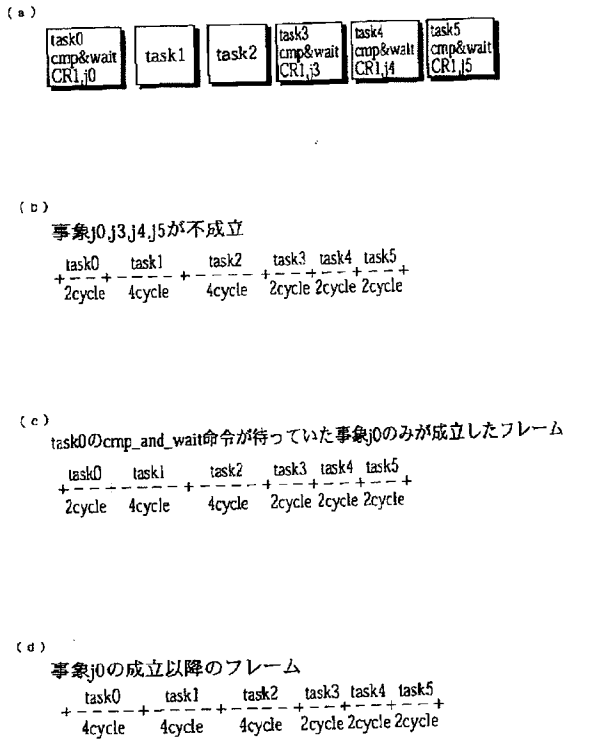
【図２１】


wait状態6bit
タスク管理レジスタ
13
61
taskID
53
55
2
4
taskend
54
比較器
タスク切り替え信号
chgtaskex
52
counter
51
countq
countd
FF

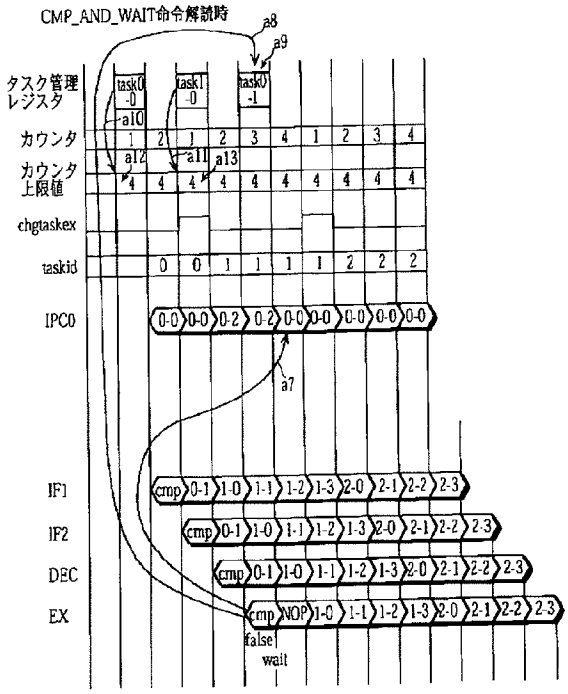
【図２３】


bit4 bit3 bit2 bit1 bit0
CR1
転送数カウントエンド
MPEGストリームからスタートコード検出
バッファに有効データが存在
バッファに有効データが3Byte以上存在
bit4 bit3 bit2 bit1 bit0
CR2
バージングタスクからDMAサポート要求発生
SDRAMにおけるDMA完了
FIFOからDMAサポート要求発生
FIFOからストリーム制御要求発生
bit4 bit3 bit2 bit1 bit0
CR3
バッファメモリがDMA_READY状態
SDRAMにおけるDMA完了
水平帰線カウンタリセット
水平ブランキング期間
他タスクからDMAサポート要求発生

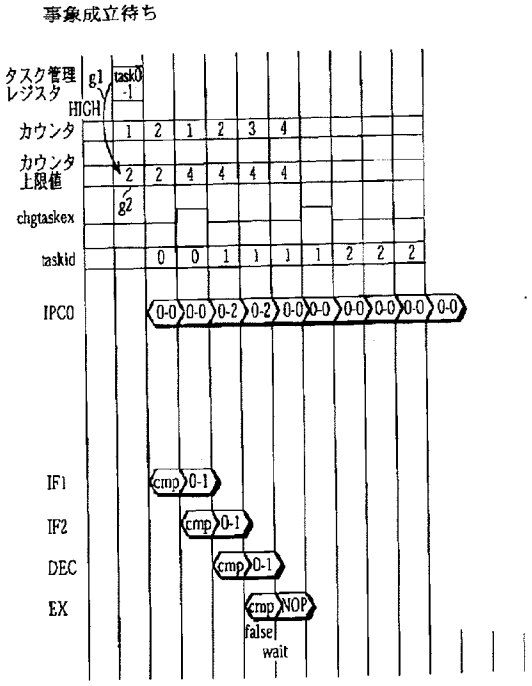
【図２４】


(a)
task0 cmp&wait CR1,j0
task1
task2
task3 cmp&wait CR1,j3
task4 cmp&wait CR1,j4
task5 cmp&wait CR1,j5
(b)
事象j0,j3,j4,j5が不成立
task0 2cycle
task1 4cycle
task2 4cycle
task3 2cycle
task4 2cycle
task5 2cycle
(c)
task0のcmp_and_wait命令が待っていた事象j0のみが成立したフレーム
task0 2cycle
task1 4cycle
task2 4cycle
task3 2cycle
task4 2cycle
task5 2cycle
(d)
事象j0の成立以降のフレーム
task0 4cycle
task1 4cycle
task2 4cycle
task3 2cycle
task4 2cycle
task5 2cycle

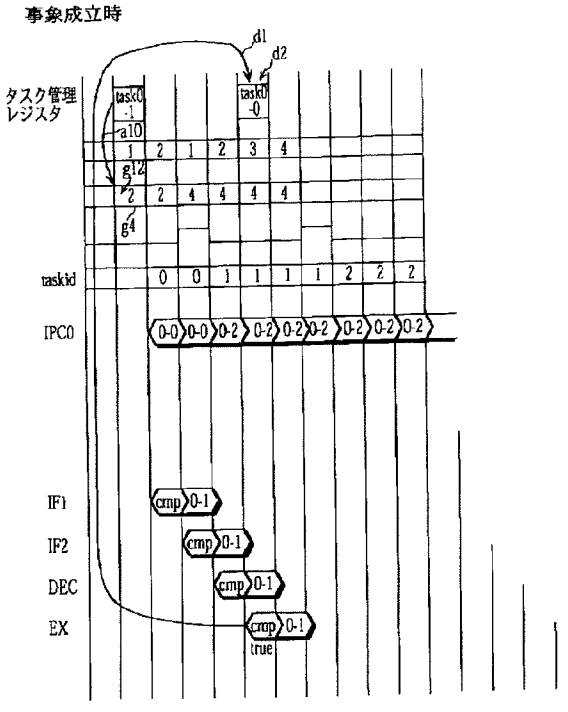
【図２６】



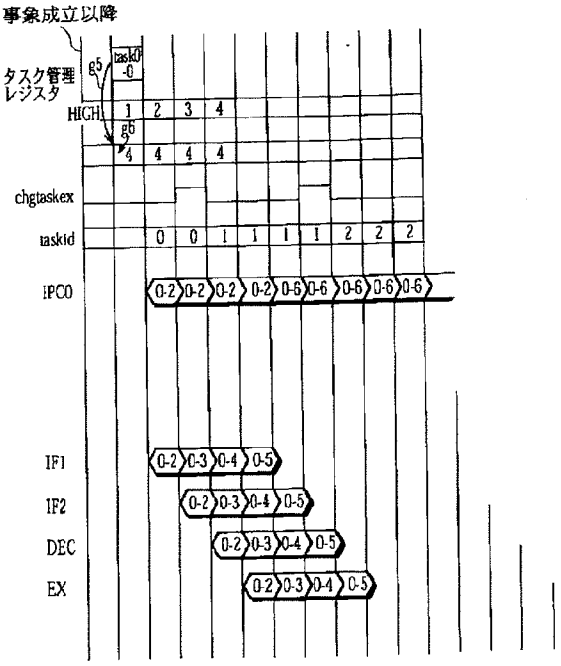
【図２７】



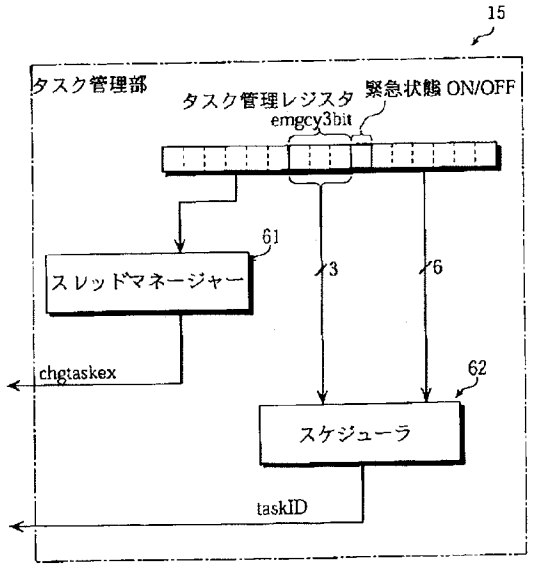
【図２８】



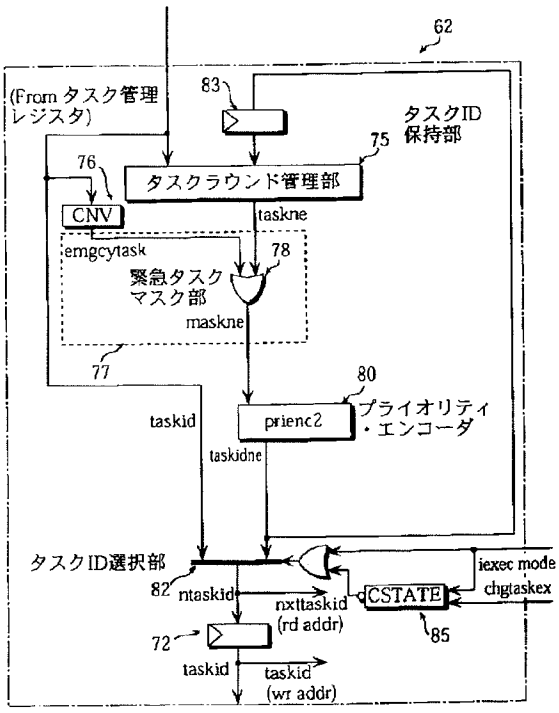
【図２９】



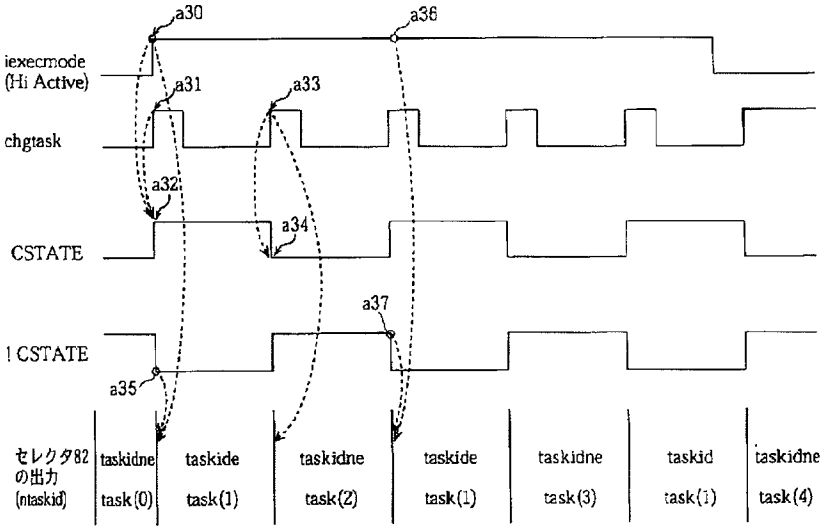
【図３０】



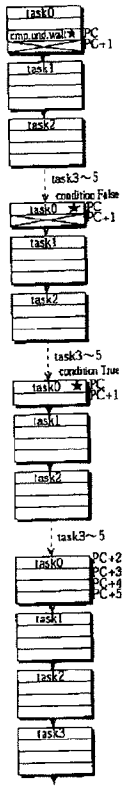
【図３１】



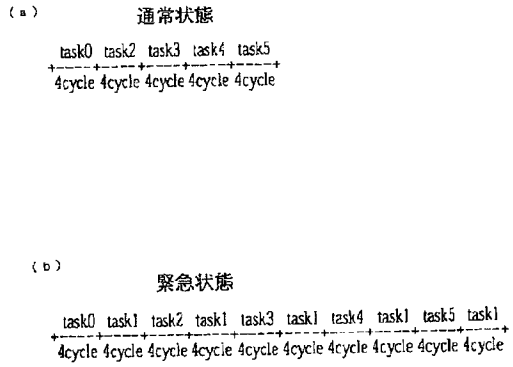
【図３４】



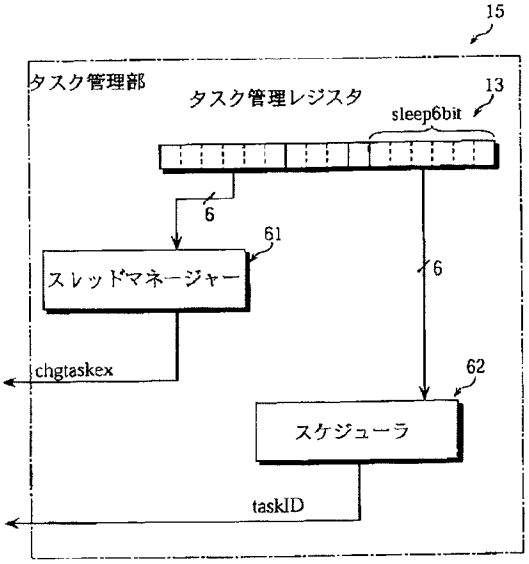
【図５２】



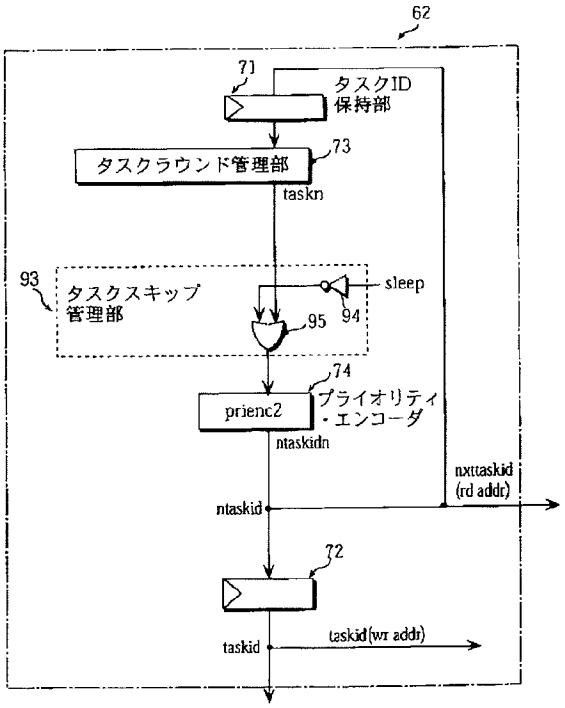
【図３５】



【図３６】



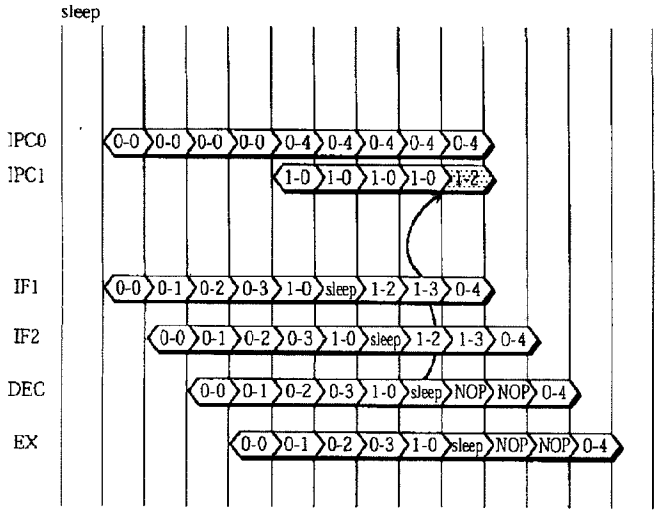
【図３８】



【図３９】

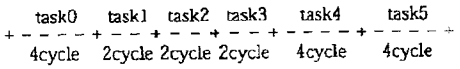
task0 メモリ制御	task1 バージング	task2	task3 HOST I/F	task4	task5
命令0-0	命令1-0	命令2-0	命令3-0	命令4-0	命令5-0
命令0-1	命令1-1 ●sleep	命令2-1 ●sleep	命令3-1 ●sleep	命令4-1	命令5-1
命令0-2	命令1-2	命令2-2	命令3-2	命令4-2	命令5-2
命令0-3	命令1-3	命令2-3	命令3-3	命令4-3	命令5-3
命令0-4	命令1-4	命令2-4	命令3-4	命令4-4	命令5-4
命令0-5	命令1-5	命令2-5	命令3-5	命令4-5	命令5-5
命令0-6	命令1-6	命令2-6	命令3-6	命令4-6	命令5-6
命令0-7	命令1-7	命令2-7	命令3-7	命令4-7	命令5-7
命令0-8	命令1-8	命令2-8	命令3-8	命令4-8	命令5-8
命令0-9	命令1-9	命令2-9	命令3-9	命令4-9	命令5-9
命令0-10 ★wake(3)	命令1-10	命令2-10	命令3-10	命令4-10	命令5-10
命令0-11	命令1-11	命令2-11	命令3-11	命令4-11	命令5-11
命令0-12	命令1-12	命令2-12	命令3-12	命令4-12	命令5-12
命令0-13 ★wake(1)	命令1-13	命令2-13	命令3-13	命令4-13	命令5-13
命令0-14	命令1-14	命令2-14	命令3-14	命令4-14	命令5-14
命令0-15	命令1-15	命令2-15	命令3-15	命令4-15	命令5-15

【図４１】

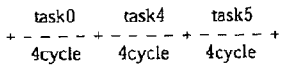


【図４２】

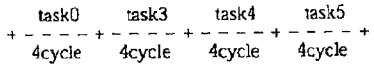
(a)
task1 task2 task3内のsleep命令の実行時



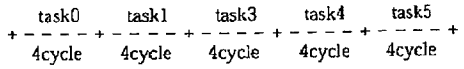
(b)
task1 task2 task3内の休眠時



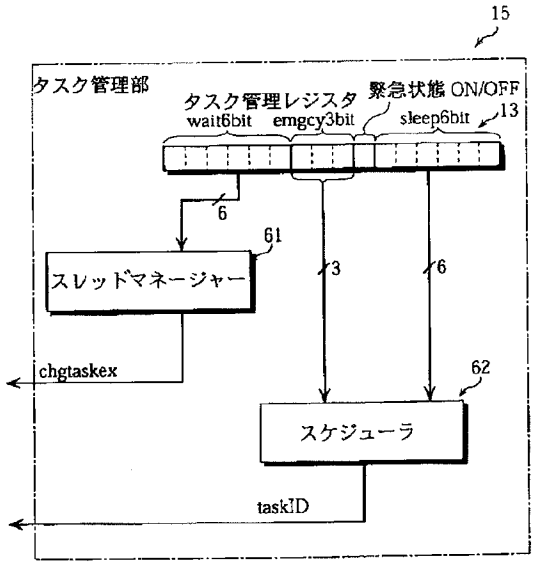
(c)
task3の起動時



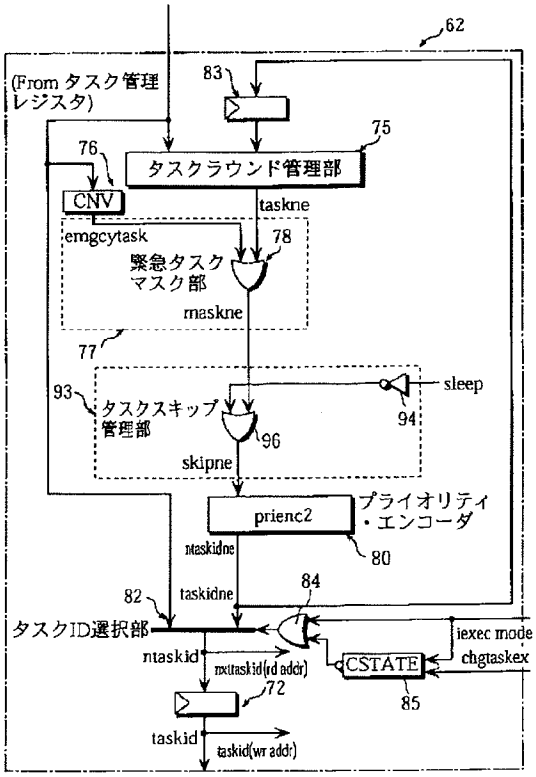
(d)
task1の起動時



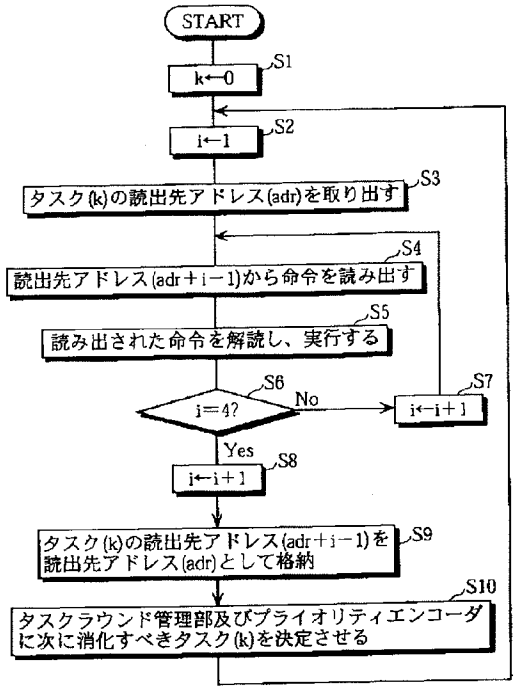
【図４３】



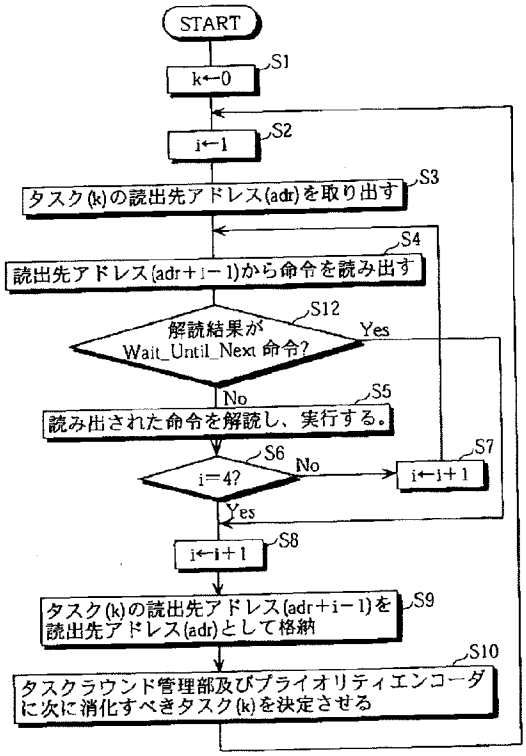
【図４４】



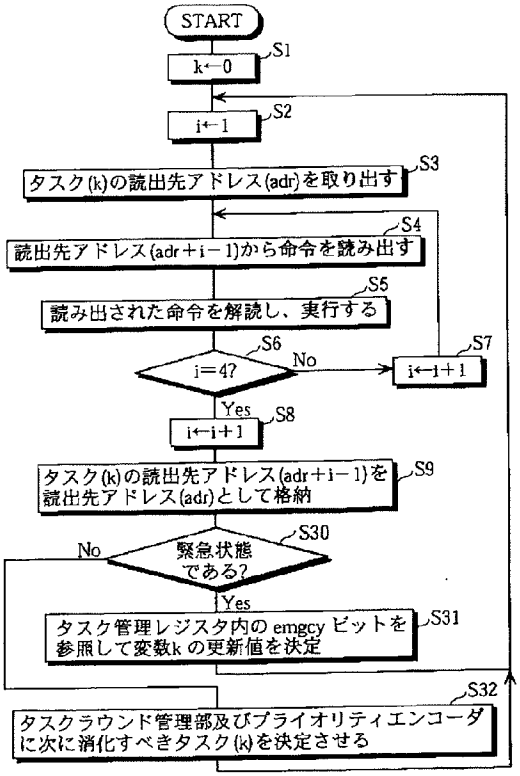
【図４５】


START
k←0
S1
i←1
S2
タスク(k)の読出先アドレス(adr)を取り出す
S3
読出先アドレス(adr+i−1)から命令を読み出す
S4
読み出された命令を解読し、実行する
S5
i=4?
S6
No
i←i+1
S7
Yes
i←i+1
S8
タスク(k)の読出先アドレス(adr+i−1)を
読出先アドレス(adr)として格納
S9
タスクラウンド管理部及びプライオリティエンコーダ
に次に消化すべきタスク(k)を決定させる
S10

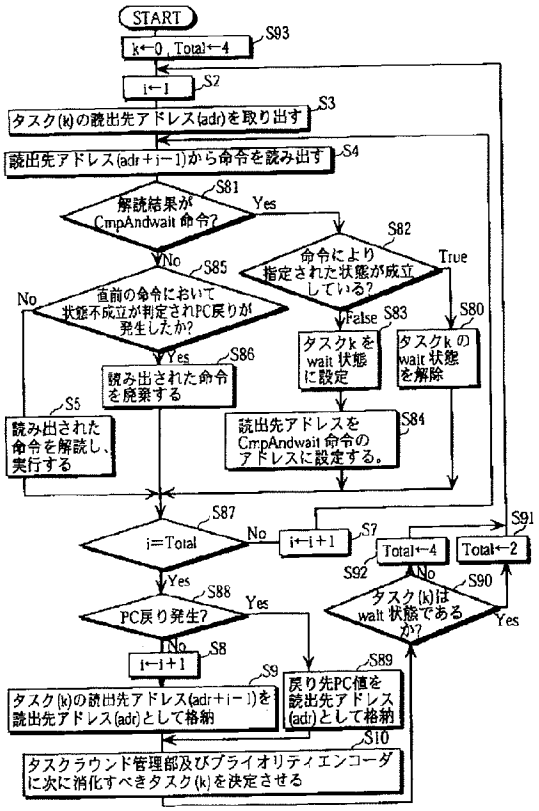
【図４６】


START
k←0
S1
i←1
S2
タスク(k)の読出先アドレス(adr)を取り出す
S3
読出先アドレス(adr+i−1)から命令を読み出す
S4
解読結果が
Wait_Until_Next 命令?
S12
Yes
No
読み出された命令を解読し、実行する。
S5
i=4?
S6
No
i←i+1
S7
Yes
i←i+1
S8
タスク(k)の読出先アドレス(adr+i−1)を
読出先アドレス(adr)として格納
S9
タスクラウンド管理部及びプライオリティエンコーダ
に次に消化すべきタスク(k)を決定させる
S10

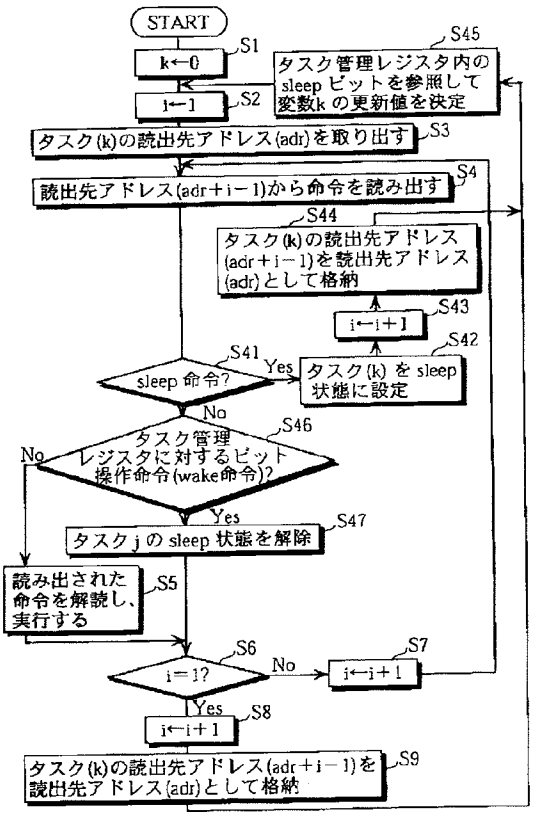
【図４８】


START
k←0
S1
i←1
S2
タスク(k)の読出先アドレス(adr)を取り出す
S3
読出先アドレス(adr+i−1)から命令を読み出す
S4
読み出された命令を解読し、実行する
S5
i=4?
S6
No
i←i+1
S7
Yes
i←i+1
S8
タスク(k)の読出先アドレス(adr+i−1)を
読出先アドレス(adr)として格納
S9
No
緊急状態
である?
S30
Yes
タスク管理レジスタ内の emgcy ビットを
参照して変数k の更新値を決定
S31
タスクラウンド管理部及びプライオリティエンコーダ
に次に消化すべきタスク(k)を決定させる
S32

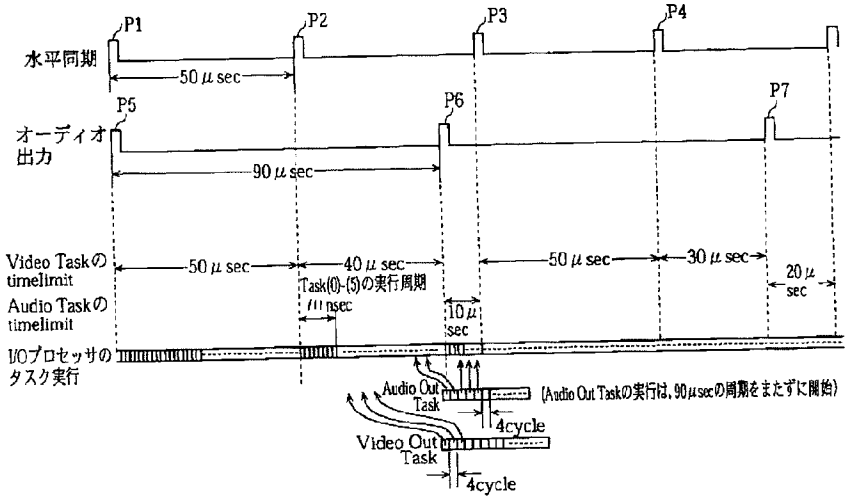
【図４７】


START
k←0, Total←4
S93
i←1
S2
タスク(k)の読出先アドレス(adr)を取り出す
S3
読出先アドレス(adr+i−1)から命令を読み出す
S4
解読結果がCmpAndwait命令?
S81
Yes
No
命令により指定された状態が成立している?
S82
True
False
S83
S80
タスクkをwait状態に設定
タスクkのwait状態を解除
直前の命令において状態不成立が判定されPC戻りが発生したか?
S85
No
Yes
S86
読み出された命令を廃棄する
S5
読み出された命令を解読し、実行する
読出先アドレスをCmpAndwait命令のアドレスに設定する。
S84
i=Total
S87
No
i←i+1
S7
Total←4
Total←2
S91
S92
No
Yes
タスク(k)はwait状態であるか?
S90
Yes
PC戻り発生?
S88
Yes
No
i←i+1
S8
S89
戻り先PC値を読出先アドレス(adr)として格納
S9
タスク(k)の読出先アドレス(adr+i−1)を読出先アドレス(adr)として格納
S10
タスクラウンド管理部及びプライオリティエンコーダに次に消化すべきタスク(k)を決定させる

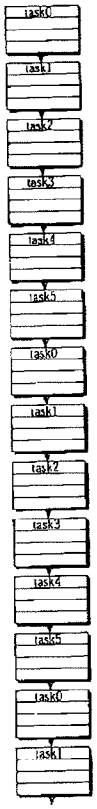
【図４９】


START
k←0
S1
S45
タスク管理レジスタ内のsleepビットを参照して変数kの更新値を決定
i←1
S2
タスク(k)の読出先アドレス(adr)を取り出す
S3
読出先アドレス(adr+i−1)から命令を読み出す
S4
S44
タスク(k)の読出先アドレス(adr+i−1)を読出先アドレス(adr)として格納
i←i+1
S43
S42
sleep命令?
S41
Yes
タスク(k)をsleep状態に設定
No
タスク管理レジスタに対するビット操作命令(wake命令)?
S46
No
Yes
タスクjのsleep状態を解除
S47
読み出された命令を解読し、実行する
S5
i=1?
S6
No
i←i+1
S7
Yes
i←i+1
S8
タスク(k)の読出先アドレス(adr+i−1)を読出先アドレス(adr)として格納
S9

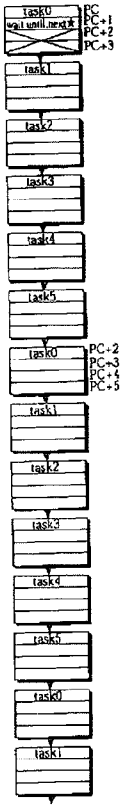
【図５５】


P1
P2
P3
P4
水平同期
50μsec
P5
P6
P7
オーディオ出力
90μsec
Video Taskのtimelimit
50μsec
40μsec
Task(0)-(5)の実行周期
111nsec
50μsec
30μsec
20μsec
Audio Taskのtimelimit
10μsec
I/Oプロセッサのタスク実行
Audio Out Task
(Audio Out Taskの実行は、90μsecの周期をまたずに開始)
4cycle
Video Out Task
4cycle

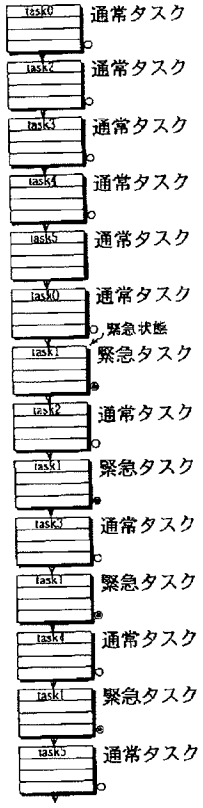
【図５０】



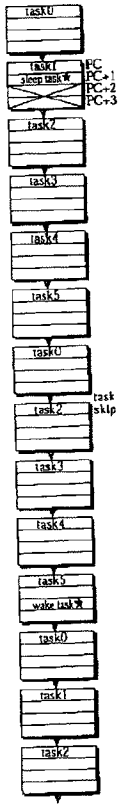
【図５１】



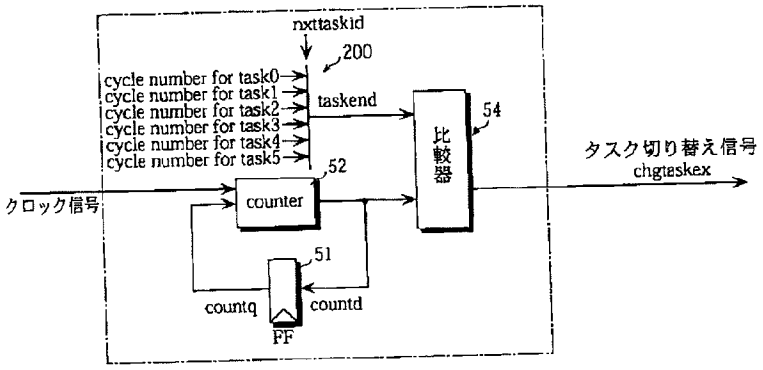
【図５３】



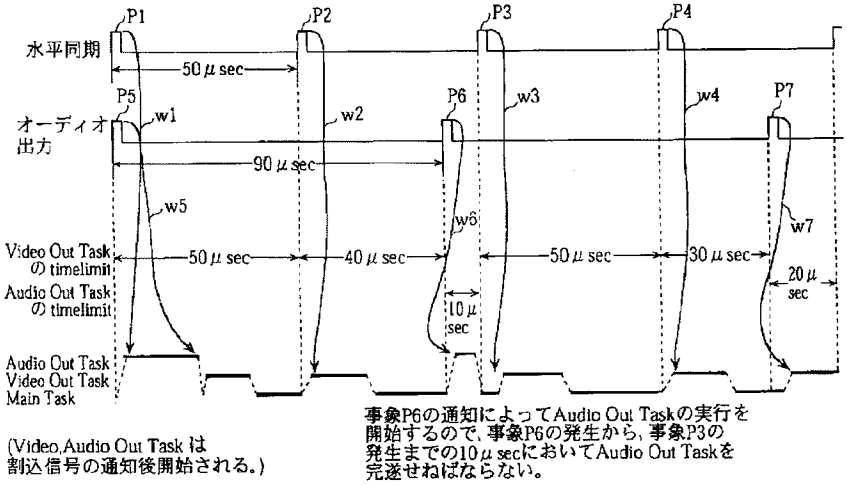
【図５４】



【図５６】



【図５７】



フロントページの続き

(72)発明者　平井　誠
大阪府門真市大字門真1006番地　松下電器産業株式会社内

(72)発明者　清原　督三
大阪府門真市大字門真1006番地　松下電器産業株式会社内